எளிய தமிழில்

து. நித்யா

கணியம் வெளியீடு

எளிய தமிழில் **MySQL**

து.நித்யா

மின்னூல் வெளியீடு **: http://FreeTamilEbooks.com**

சென்னை

எளிய தமிழில் MySQL Copyright © 2014 by Creative Commons Attribution-ShareAlike 4.0 International License.

This book was produced using PressBooks.com.

Contents

- எளிய தமிழில் MySQL
- உரிமை
- நன்கொடை
- 1. MySQL - ஓர் அறிமுகம்
- 2. SQL Command line client-ஐ பயன்படுத்துதல்
- 3. Databases பற்றிய அறிமுகம்
- 4. Tables பற்றிய அறிமுகம்
- 5. Columns - அறிமுகம்
- 6. Indexes - அறிமுகம்
- 7. Library எனும் 'மாதிரி Databases' -ஐ அமைத்தல்
- 8. Library எனும் மாதிரி Database-ல் data-வை செலுத்துதல்
- 9. Library எனும் மாதிரி Database-ல் இருக்கும் data-வை எடுத்தல்
- 10. Library database-ல் இருக்கும் பல்வேறு Tables-ஐ இணைத்தல்
- 11. Library எனும் database-ல் சேமிக்கப்பட்டிருக்கும் data-ஐ மாற்றுதல்
- 12. Library எனும் database-ல் சேமிக்கப்பட்டிருக்கும் data-ஐ நீக்குதல்
- 13. Users-ஐ கையாளுதல்
- 14. MySQL-ன் Application Programming Interface - API
- 15. Backup எடுத்தல் மற்றும் Trouble Shooting செய்தல்
- Free Tamil Ebooks - எங்களைப் பற்றி

எளிய தமிழில் **MySQL**

MySQL பரவலாக பயன்படுத்தப்படும் ஒரு கட்டற்ற மென்பொருள்
(Free Open Source Software) வகையிலான Database System.
இதை , இந்த நூல் எளிமையாக அறிமுகம் செய்கிறது .

தமிழில் கட்டற்ற மென்பொருட்கள்
பற்றிய தகவல்களை “ கணியம் ” மின் மாத இதழ் , 2012
முதல் வெளியிட்டு வருகிறது . இதில் வெளியான
MySQL பற்றிய கட்டுரைகளுடன் , மேலும் புதிய
பகுதிகளை இணைத்து ஒரு முழு புத்தகமாக

வெளியிடுவதில் பெரு மகிழ்ச்சி கொள்கிறோம் .

உங்கள் கருத்துகளையும் , பிழை திருத்தங்களையும் editor@kaniyam.com க்கு மின்னஞ்சல் அனுப்பலாம் .

http://kaniyam.com/mysql-book-in-tamil என்ற முகவரியில் இருந்து இந்த நூலை பதிவிறக்கம் செய்யலாம் . உங்கள் கருத்துகளையும் இங்கே பகிரலாம் .

படித்து பயன் பெறவும் , பிறருடன் பகிர்ந்து மகிழவும் வேண்டுகிறோம் .

கணியம் இதழை தொடர்ந்து வளர்க்கும் அனைத்து அன்பர்களுக்கும் எமது நன்றிகள் .

து.நித்யா

nithyadurai87@gmail.com

மின்னூலாக்கம் : து.நித்யா

மின்னூல் வெளியீடு : FreeTamilEbooks.com

உரிமை – கிரியேட்டிவ் காமன்ஸ். எல்லாரும் படிக்கலாம், பகிரலாம்.

Creative Commons Attribution-ShareAlike 4.0 International License.

உரிமை

இந்த நூல் கிரியேடிவ் காமன்ஸ் என்ற உரிமையில் வெளியிடப்படுகிறது . இதன் மூலம் , நீங்கள்

- யாருடனும் பகிர்ந்து கொள்ளலாம் .
- திருத்தி எழுதி வெளியிடலாம் .
- வணிக ரீதியிலும்யன்படுத்தலாம் .

ஆனால் , மூலப் புத்தகம் , ஆசிரியர் மற்றும் www.kaniyam.com பற்றிய விவரங்களை சேர்த்து தர வேண்டும் . இதே உரிமைகளை யாவருக்கும் தர வேண்டும் . கிரியேடிவ் காமன்ஸ் என்ற உரிமையில் வெளியிட வேண்டும் .

Creative Commons Attribution-ShareAlike 4.0 International License.

3

நன்கொடை

Creative Commons உரிமையில் தமது நூல்களை வெளியிடும் எழுத்தாளரை உங்கள் நன்கொடைகள் ஊக்குவிக்கும்.

வங்கி விவரங்கள்.

Name – Nithya Duraisamy
ICICI – 006101540799
Branch – Mcity branch,
chengalpattu.
IFSC code – ICIC0000061

THIS BOOK WAS PRODUCED USING

PRESSBOOKS.COM

Easily turn your manuscript into

EPUB *Nook, Kobo, and iBooks*

Mobi *Kindle*

PDF *Print-on-demand and digital distribution*

PRESSBOOKS.COM

Simple Book Production

1

MySQL - ஓர் அறிமுகம்

Database என்பது நமக்கு வேண்டிய data-வை எல்லாம் ஓர் ஒழுங்குபடுத்தப்பட்ட, கட்டுக்கோப்பான வடிவில் சேமிக்க உதவும் ஒரு சாப்ட்வேர் ஆகும். SQL(Structured Query Language) என்பது database-ல் data-வை கையாளுவதற்கு நாம் பயன்படுத்தும் ஒரு கணினி மொழி ஆகும். RDBMS என்பது, ஒரு database-இல் பல்வேறு இடங்களில் சேமிக்கப்பட்டிருக்கும் data-வை ஒன்றுடன் ஒன்று தொடர்பு படுத்தி அதனை மொத்தமாக நிர்வாகம் செய்ய உதவும் ஒரு Management software ஆகும்.

MySQL என்பது இவ்வகையான ஒரு RDBMS Software ஆகும். இது SQL மொழியில் மட்டும் அல்லாமல் PHP, PERL, C, C++, JAVA போன்ற பல்வேறு கணினி மொழிகளிலும் இயங்கவல்லது. இது free software வகையைச் சேர்ந்தது மற்றும் GPL எனப்படும் General Public License-வுடன் வரக்கூடியது. எனவே நாம் இதனை எந்தவித கட்டணமும் இன்றி Internet-ல் இருந்து இலவசமாக download செய்து நமது தேவைக்காகப் பயன்படுத்தலாம்.

1.1 MySQL – இன் வடிவமைப்பு

MySQL-ஆனது பற்பல கூறுகளை உள்ளடக்கிய ஒரு மென்பொருள் அமைப்பாகும். பொதுவாக இதனை MySQL server மற்றும் MySQL client என இரண்டு வகையாகப் பிரிக்கலாம். MySQL client என்பது நம்மால் பார்த்து பயன்படுத்தக் கூடிய வகையில் இருக்கும் ஒரு front end tool ஆகும். அதாவது Windows-ல் இருக்கும் console prompt மற்றும் GNU/Linux–ல் இருக்கும் shell prompt போன்றவற்றின் மூலமாக, நாம் SQL மொழியில் commands வழங்கலாம். இந்த commands-ஐத் தான் MySQL server பெற்றுக்கொண்டு அதற்குரிய வேலைகளைச் செய்யத் துவங்கும். MySQL server-ல் என்ன நிகழ்கிறது என்பது

பொதுவாக பயனர்களின் கண்களுக்குப் புலப்படாது. ஆனால் இந்த MySQL server-தான் எல்லா வேலைகளையும் செய்து முடித்து result-ஐக் கொடுக்கும்.

இந்த வரைபடத்தில் பல்வேறு MySQL clients ஆனது ஒரு MySQL Server வுடன் இணைக்கப்பட்டுள்ளது.

ஒவ்வொரு MySQL client-வும் பின்வரும் வேலைகளைப் புரிகிறது.

- Password ஐ சரிபார்த்து Authentication ஐ தொடங்குதல்
- நாம் கொடுக்கும் SQL Queries ஐ server-க்கான Tokens ஆக மாற்றுதல்
- Tokens-ஐ Server-க்கு வழங்குதல்
- Compress அல்லது Encrypt செய்யப்பட்ட இணைப்புகளை கண்காணித்தல்
- கடைசியாக Server-இடம் இருந்து விடைகளைப் பெற்றுக்கொண்டு அதனை பயனர்களுக்குத் தெரிவித்தல்

MySQL server ஆனது client இடமிருந்து request-ஐ பெற்றுக்கொண்டு அதற்குரிய response-ஐ திரும்பக்கொடுக்கும்.இது Management Layer மற்றும் Storage Engine என்று இரண்டு பெரும் பகுதிகளை உள்ளடக்கியுள்ளது .இவைதான் அதிக அளவில் memory, disk மற்றும் network- வுடன் தொடர்பு கொள்கின்றன.

Management Layer ஆனது, MySQL client இடமிருந்து பெரும் request-ஐ வைத்துக்கொண்டு, பின்வரும் வேலைகளைப் புரிகிறது.

- இணைப்புகளை decrypt அல்லது decode செய்தல்
- Queries ஐ சரிபார்த்து parse செய்தல்

- Query Cache -லிருந்து catched queries -ஐ எடுத்தல்
- தகவல்களை Storage Engine-க்கு அனுப்புதல்

மேலும் disk-க்கான logs-ஐ எழுதுதல் மற்றும் memory-ல் logs-ஐ சேமித்தல் மற்றும் எடுத்தல் போன்ற சில வேலைகளையும் செய்கிறது.

Storage Engine ஆனது databases, tables, indexes -ஐ நிர்வகிக்கின்றது. மேலும் ஒருசில logs மற்றும் சில புள்ளிவிவரங்களையும் நிர்வகிக்கின்றது. இது இவ்வகையான data- வை disk மற்றும் memory-ல் சேமிக்கிறது. மேலும் இதனை Network மூலமாக தொலைவில் உள்ள வேறுசில MySQL server-க்கு அனுப்புதல் போன்ற சில வேலைகளையும் செய்கிறது.

1.2 MySQL-ஐ install செய்தல்

MySQL-ஐ install செய்யும்போது, நாம் கவனிக்க வேண்டிய விஷயங்கள் பின்வருமாறு.

- MySQL-ஐ install செய்வதற்கு நமக்கு அனுமதி இருக்க வேண்டும். பொதுவாக MySQL-ன் ஒரு பிரதியை நமது machine-ல் install செய்து கொள்வது நல்லது. இதை மிகவும் கடினமான விஷயமாக எண்ணி வருந்தத் தேவையில்லை. ஏனெனில் BSD,LinuX, Mac OS X, Solaris மற்றும் windows போன்ற பெரும்பாலான platform-ல் இந்த MySQL-ஆனது எளிதாக install செய்யப்படும்.
- MySQL server-ஐ அணுகுவதற்கு அனைத்து permissions-ஐயும் கொண்ட ஒரு account இருக்க வேண்டும். பொதுவாக MySQL-ஐ முதன்முதலில் install செய்யும்போது உருவாக்கப்படும் root account இவ்வாறு அனைத்து permissions-ஐயும்

கொண்டதாக அமையும்.
- பின்னர் ஒரு MySQL client தேவை. பொதுவாக MySQL command-line அல்லது MySQL query browser ஆகியவற்றைப் பயன்படுத்துவது சிறந்தது. இந்தப் பகுதியில் உள்ள அனைத்து உதாரணங்களும் MySQL command-line-ஐக் கருத்தில் கொண்டே கொடுக்கப்பட்டுள்ளன.

1.3 Ubuntu Linux-ல் MySQL-ஐ install செய்தல்

சமீபத்திய உபுண்டு லினக்ஸ் 12.10-ல் MySQL நிறுவும் வழிகளை இங்கு காண்போம். இது debian மற்றும் ubuntu சார்ந்த Linux Mint distribution-களுக்கும் பொருந்தும்.

Terminal-ல் கீழ்வரும் கட்டளையைத் தரவும்.

sudo apt-get install mysql-server mysql-client
இப்போது MySQL-க்கு தேவையான மென்பொருட்கள் அனைத்தும் repository-ல் இருந்து download ஆகி install செய்யப்படும். இப்போது MySQL-ன் root user-க்கான password கேட்கப்படும். மிகவும் சிக்கலான சொல்லையே password ஆக தர வேண்டும்.

MySQL-ன் செயல்பாடுகளை மேலும் பாதுகாக்க கீழ்வரும் கட்டளையை இயக்கவும்.

sudo mysql_secure_installation

இது பின்வரும் பணிகளைச் செய்கிறது.

1. Anonymous users-ஐ நீக்குதல்
2. localhost-ல் இருந்து மட்டுமே root-ஐ அனுமதித்தல்
3. test database-ஐ நீக்குதல்

```
By default, a MySQL installation has an anonymous user,
to log into MySQL without having to have a user account
them.  This is intended only for testing, and to make th
go a bit smoother.  You should remove them before moving
```

```
production environment.
Remove anonymous users? [Y/n] y
 ... Success!
Normally, root should only be allowed to connect from 'l
ensures that someone cannot guess at the root password f
Disallow root login remotely? [Y/n] y
 ... Success!
By default, MySQL comes with a database named 'test' tha
access.  This is also intended only for testing, and sho
before moving into a production environment.
Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!
Reloading the privilege tables will ensure that all char
will take effect immediately.
Reload privilege tables now? [Y/n] y
 ... Success!
Cleaning up...
All done!  If you've completed all of the above steps, y
installation should now be secure.
Thanks for using MySQL!
```

அவ்வளவுதான் MySQL-ஐ நிறுவிவிட்டோம்.

1.4 Configuration

இப்போது MySQL-ன் பல்வேறு configuration options-ஐப் பற்றிப் பார்ப்போம்.

/etc/mysql என்ற directory-தான் MySQL-ன் config directory ஆகும்.

/etc/mysql/my.cnf

என்ற file ஆனது MySQL-ன் மொத்த configuration options-ஐயும் கொண்டுள்ளது.

Log file, port number, ip binding, performance என பல்வேறு

options உள்ளன. இதில் சில options பற்றி இங்கே பார்ப்போம்.

port = 3306

இது MySQL server-ஆனது 3306 என்ற port-ல் இயங்குவதைக் குறிக்கிறது.

user = mysql

லினக்ஸில் mysql என்ற user உருவாகி, அதே user-ஆக server இயங்கும். இதனால் லினக்ஸ் server-ன் பாதுகாப்பு அதிகரிக்கிறது.

data dir = /var/lib/mysql

MySQL-ன் database அனைத்தும் இந்த folder-ல் சேமிக்கப்படுகின்றன. Backup செய்யும் போது, இந்த folder-ஐ கண்டிப்பாக backup செய்ய வேண்டும்.

bind-address = 127.0.0.1

எந்த ip address-ல் MySQL server இயங்க வேண்டும் என இங்கே தரலாம்.

log_error = /var/log/mysql/error.log

Configuration அல்லது connection-ல் பிழை இருந்தால் இந்த file-ல் log செய்யப்படுகிறது.

1.5 Query cache

MySQL server-ல் அடிக்கடி இயங்கும் select queries-ம், results-ம் cache-ல் சேமித்து வைக்கப்படும். இதனால், server-ன் திறனும் வேகமும் அதிகரிக்கிறது. இதற்கான options பின்வருமாறு.

query_cache_limit = 1m

query_cache_size = 16m

இங்கு m என்பது mb ஆகும். நமது RAM-ன் அளவைப்

பொறுத்து இதை அதிகரிக்கலாம். உதாரணம்.

query_cache_limit = 2m

query_cache_size = 32m

இந்த my.cnf file-ல் எந்த மாற்றம் செய்தாலும், MySQL server-ஐ restart செய்ய வேண்டும்.

MySQL server- ஐ **restart** செய்தல்

sudo service mysql restart

MySQL server- ஐ **stop** செய்தல்

sudo service mysql stop

MySQL server- ஐ **start** செய்தல்

sudo service mysql start

1.6 MySQL clients

Command line client மட்டுமின்றி GUI வழியாக, MySQL-ஐ பயன்படுத்த பல்வேறு clients உபுண்டு லினக்ஸில் உள்ளன.

MySQL Work Bench

sudo apt-get install MySQL-workbench

MySQL Navigator

sudo apt-get install MySQL-navigator

Emma

sudo apt-get install emma

MySQL Admin

sudo apt-get install MySQL-admin

PHPMyAdmin

sudo aptitude install phpmyadmin

1.7 CENTOS/RHEL – ல் நிறுவுதல்

Fedoro, CentOS மற்றும் RHEL லினக்ஸ் போன்ற distribution-நிறுவ, பின்வரும் கட்டளையை இயக்கவும்.

yum install mysql mysql-server

இதில் root-க்கு password இருக்காது. ஆனால் இது பாதுகாப்பானது அல்ல.

/usr/bin/mysqladmin -u root password 'COMPLEX-PASSWORD-HERE'

என்ற கட்டளை மூலம் சிக்கலான ஒரு password-ஐ root-க்கு தந்து பாதுகாப்பை அதிகரிக்கலாம்.

/usr/bin/mysql-secure-installation

மூலம் மேலும் பாதுகாப்பு தரலாம்.

1.8 Windows-ல் MySQL-ஐ install செய்தல்

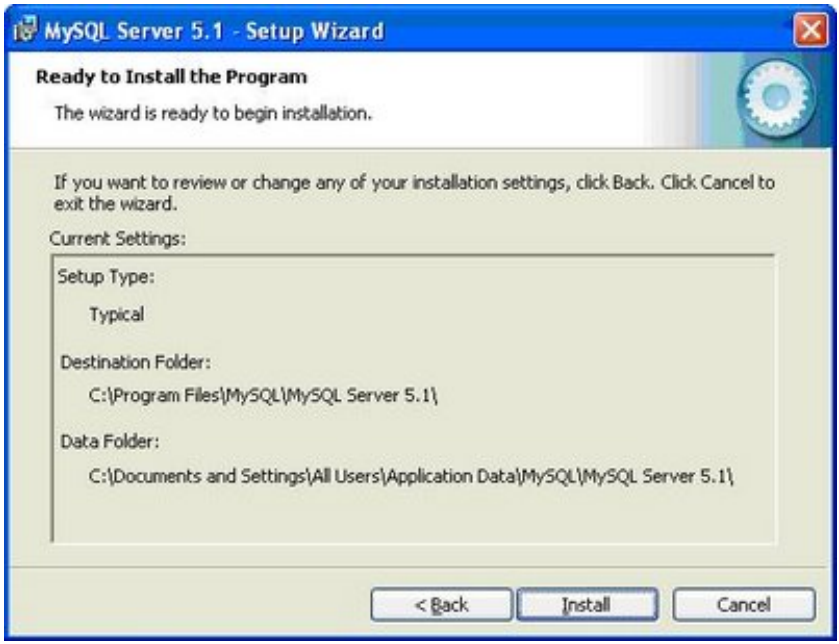
MySQL-ன் தற்போதைய பதிப்பை (MySQL Community Server 5.5.29) http://mysql.com/downloads -லிருந்து பதிவிறக்கம் செய்து கொள்ளவும். msi file-ஐ இயக்கவும்.

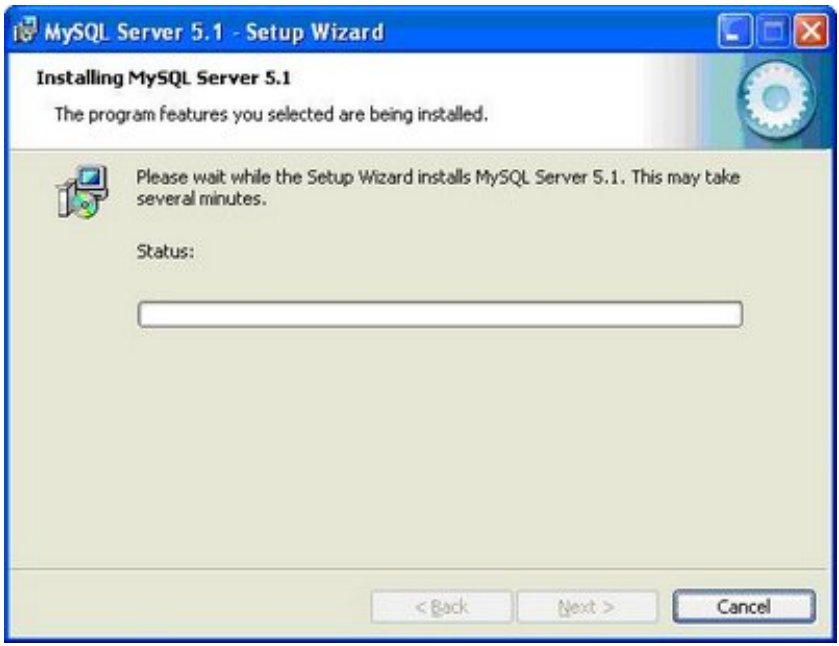
பின்வரும் திரைப்பிடிப்புகள் MySQL 5.1.35 -க்கானவை.

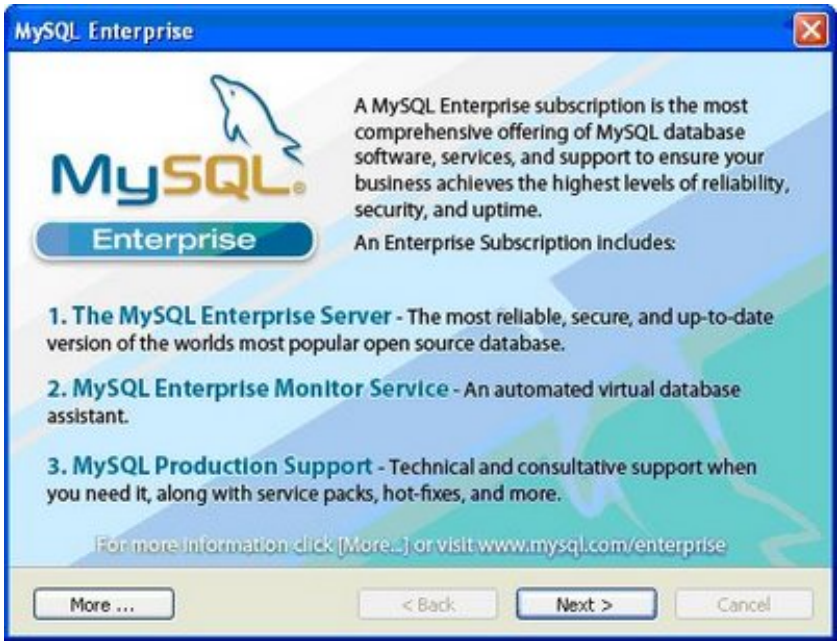
—

MySQL Server 5.1 - Setup Wizard
Welcome to the Setup Wizard for MySQL Server 5.1
The Setup Wizard will install MySQL Server 5.1 release 5.1.35 on your computer. To continue, click Next.
WARNING: This program is protected by copyright law.
MySQL
< Back
Next >
Cancel

MySQL Server 5.1 - Setup Wizard
Setup Type
Choose the setup type that best suits your needs.
Please select a setup type.
Typical
Common program features will be installed. Recommended for general use.
Complete
All program features will be installed. (Requires the most disk space.)
Custom
Choose which program features you want installed and where they will be installed. Recommended for advanced users.
< Back
Next >
Cancel

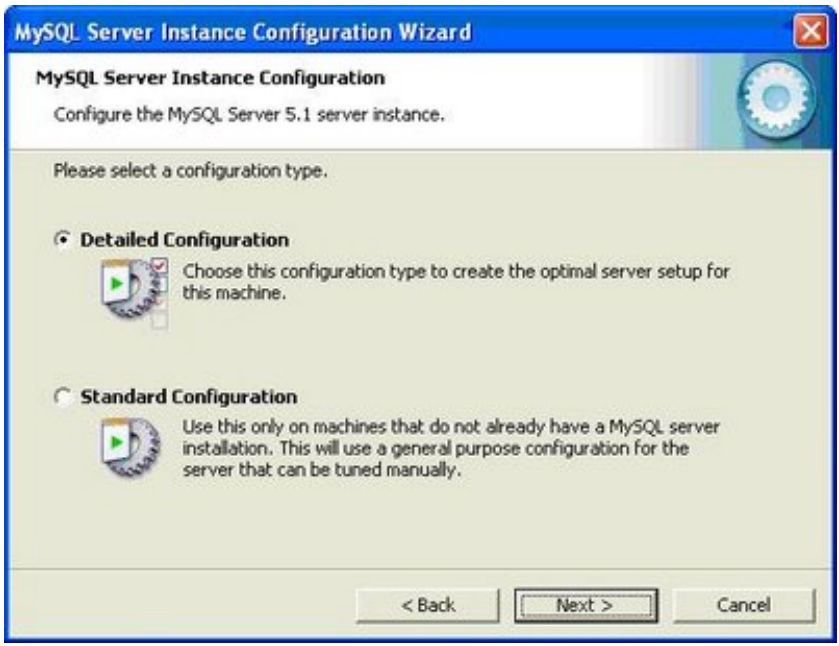

MySQL Server 5.1 - Setup Wizard
Ready to Install the Program
The wizard is ready to begin installation.
If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.
Current Settings:
Setup Type:
Typical
Destination Folder:
C:\Program Files\MySQL\MySQL Server 5.1\
Data Folder:
C:\Documents and Settings\All Users\Application Data\MySQL\MySQL Server 5.1\
< Back
Install
Cancel


MySQL Server 5.1 - Setup Wizard
Installing MySQL Server 5.1
The program features you selected are being installed.
Please wait while the Setup Wizard installs MySQL Server 5.1. This may take several minutes.
Status:
< Back
Next >
Cancel

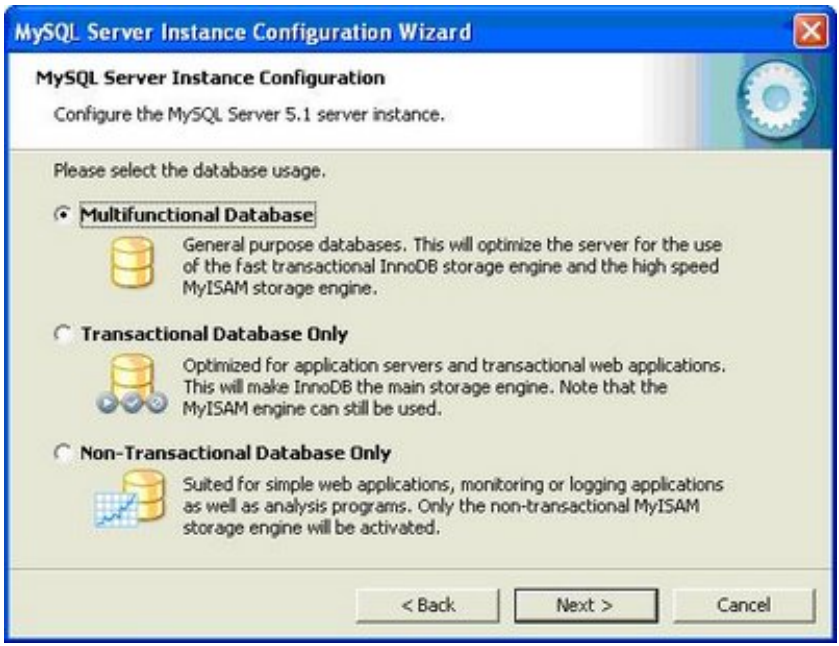

MySQL Enterprise
MySQL
Enterprise
A MySQL Enterprise subscription is the most comprehensive offering of MySQL database software, services, and support to ensure your business achieves the highest levels of reliability, security, and uptime.
An Enterprise Subscription includes:
1. The MySQL Enterprise Server - The most reliable, secure, and up-to-date version of the worlds most popular open source database.
2. MySQL Enterprise Monitor Service - An automated virtual database assistant.
3. MySQL Production Support - Technical and consultative support when you need it, along with service packs, hot-fixes, and more.
For more information click [More...] or visit www.mysql.com/enterprise
More ...
< Back
Next >
Cancel

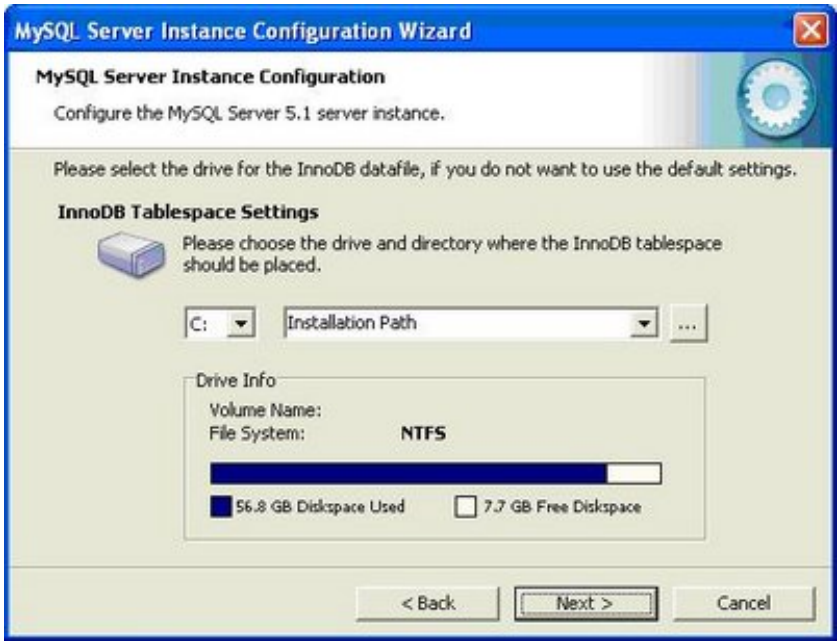
MySQL Enterprise
MySQL
Enterprise
The MySQL Enterprise Monitor Service
• Quickly identifies your most expensive SQL code across all your servers.
• MySQL Advisors and 125+ Best Practice Rules ensure security and performance.
• Alerts and Expert Advice on how to fix problems and tune for peak performance.
For more information click [More...] or visit www.mysql.com/enterprise
More ...
< Back
Next >
Cancel

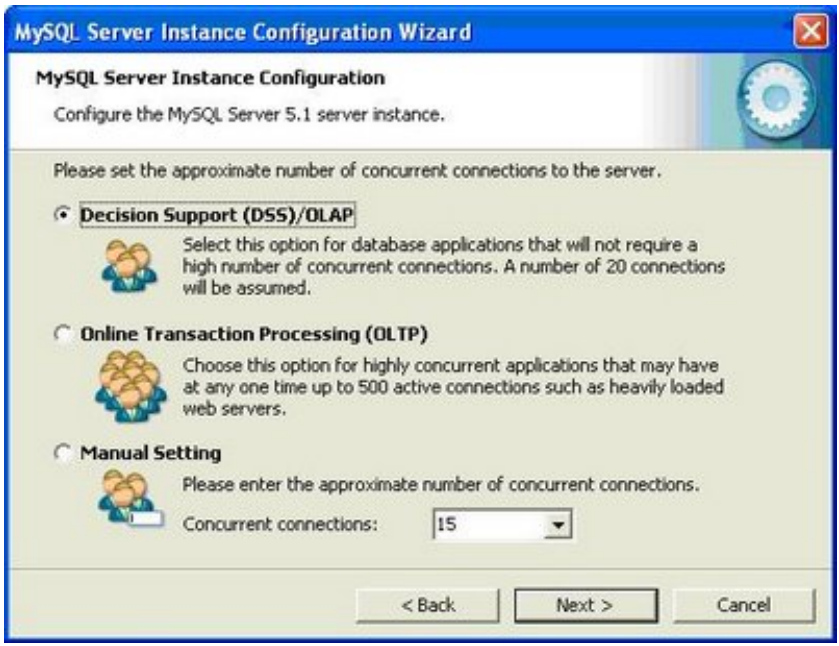
MySQL Server 5.1 - Setup Wizard
Wizard Completed
Setup has finished installing MySQL Server 5.1. Click Finish to exit the wizard.
Configure the MySQL Server now
Use this option to generate an optimized MySQL config file, setup a Windows service running on a dedicated port and to set the password for the root account.
Register the MySQL Server now
Use this option to register this MySQL server with SunConnect service, to receive automatic update notifications on future releases and other free offerings. A browser window will open briefly to allow you to complete the registration.
MySQL
< Back
Finish
Cancel

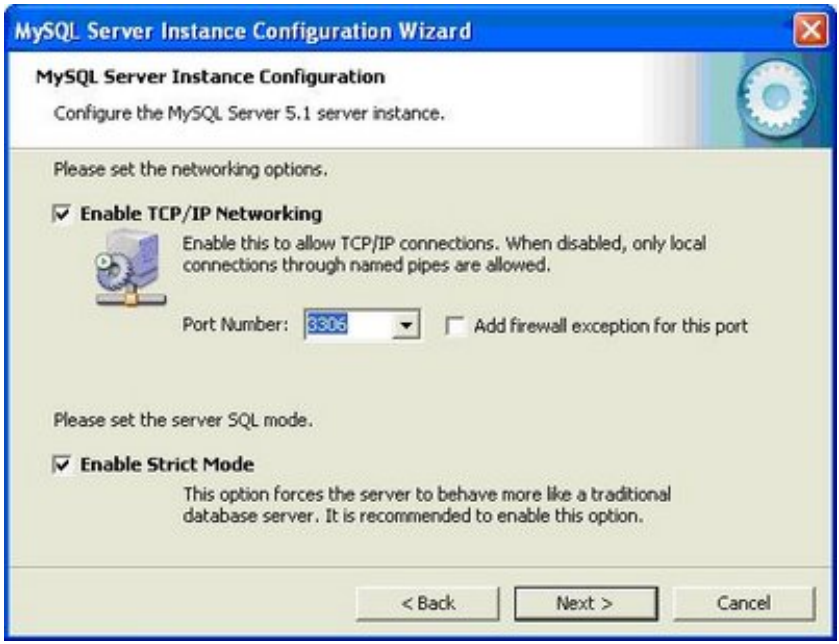

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please select a configuration type.
Detailed Configuration
Choose this configuration type to create the optimal server setup for this machine.
Standard Configuration
Use this only on machines that do not already have a MySQL server installation. This will use a general purpose configuration for the server that can be tuned manually.
< Back
Next >
Cancel

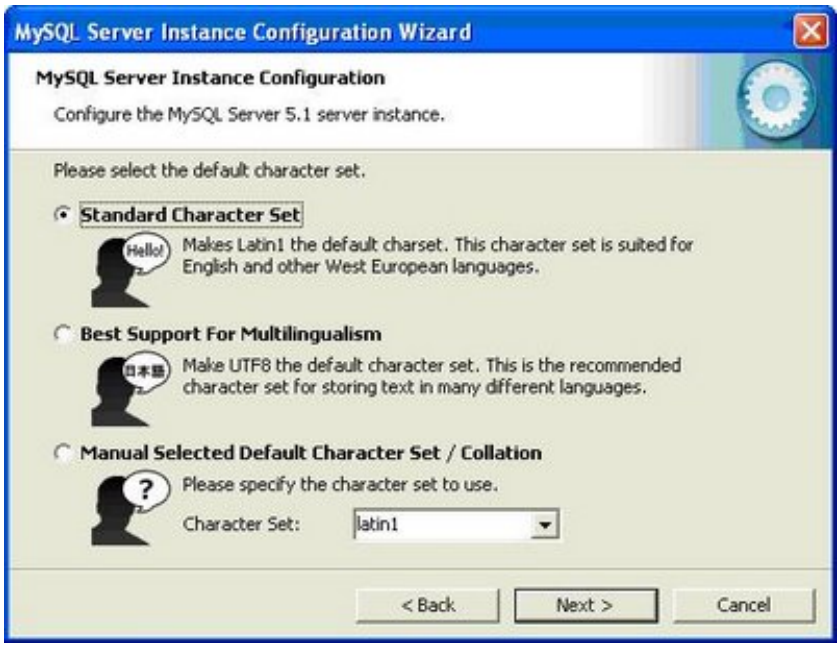
MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please select a server type. This will influence memory, disk and CPU usage.
Developer Machine
This is a development machine, and many other applications will be run on it. MySQL Server should only use a minimal amount of memory.
Server Machine
Several server applications will be running on this machine. Choose this option for web/application servers. MySQL will have medium memory usage.
Dedicated MySQL Server Machine
This machine is dedicated to run the MySQL Database Server. No other servers, such as a web or mail server, will be run. MySQL will utilize up to all available memory.
< Back
Next >
Cancel


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please select the database usage.
Multifunctional Database
General purpose databases. This will optimize the server for the use of the fast transactional InnoDB storage engine and the high speed MyISAM storage engine.
Transactional Database Only
Optimized for application servers and transactional web applications. This will make InnoDB the main storage engine. Note that the MyISAM engine can still be used.
Non-Transactional Database Only
Suited for simple web applications, monitoring or logging applications as well as analysis programs. Only the non-transactional MyISAM storage engine will be activated.
< Back
Next >
Cancel


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please select the drive for the InnoDB datafile, if you do not want to use the default settings.
InnoDB Tablespace Settings
Please choose the drive and directory where the InnoDB tablespace should be placed.
C:
Installation Path
...
Drive Info
Volume Name:
File System:
NTFS
56.8 GB Diskspace Used
7.7 GB Free Diskspace
< Back
Next >
Cancel


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please set the approximate number of concurrent connections to the server.
Decision Support (DSS)/OLAP
Select this option for database applications that will not require a high number of concurrent connections. A number of 20 connections will be assumed.
Online Transaction Processing (OLTP)
Choose this option for highly concurrent applications that may have at any one time up to 500 active connections such as heavily loaded web servers.
Manual Setting
Please enter the approximate number of concurrent connections.
Concurrent connections:
15
< Back
Next >
Cancel

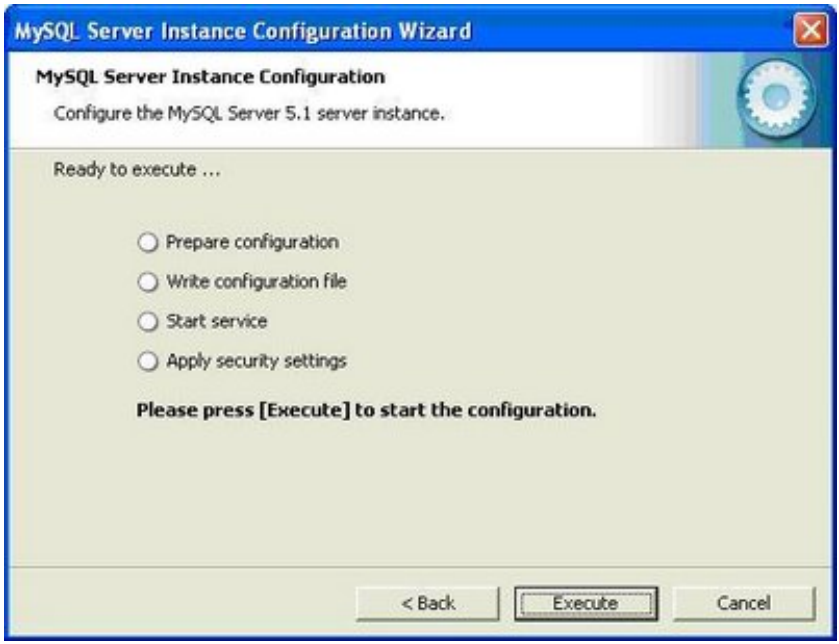

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please set the networking options.
Enable TCP/IP Networking
Enable this to allow TCP/IP connections. When disabled, only local connections through named pipes are allowed.
Port Number: 3306
Add firewall exception for this port
Please set the server SQL mode.
Enable Strict Mode
This option forces the server to behave more like a traditional database server. It is recommended to enable this option.
< Back
Next >
Cancel


MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please select the default character set.
Standard Character Set
Hello!
Makes Latin1 the default charset. This character set is suited for English and other West European languages.
Best Support For Multilingualism
Make UTF8 the default character set. This is the recommended character set for storing text in many different languages.
Manual Selected Default Character Set / Collation
Please specify the character set to use.
Character Set: latin1
< Back
Next >
Cancel

MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration

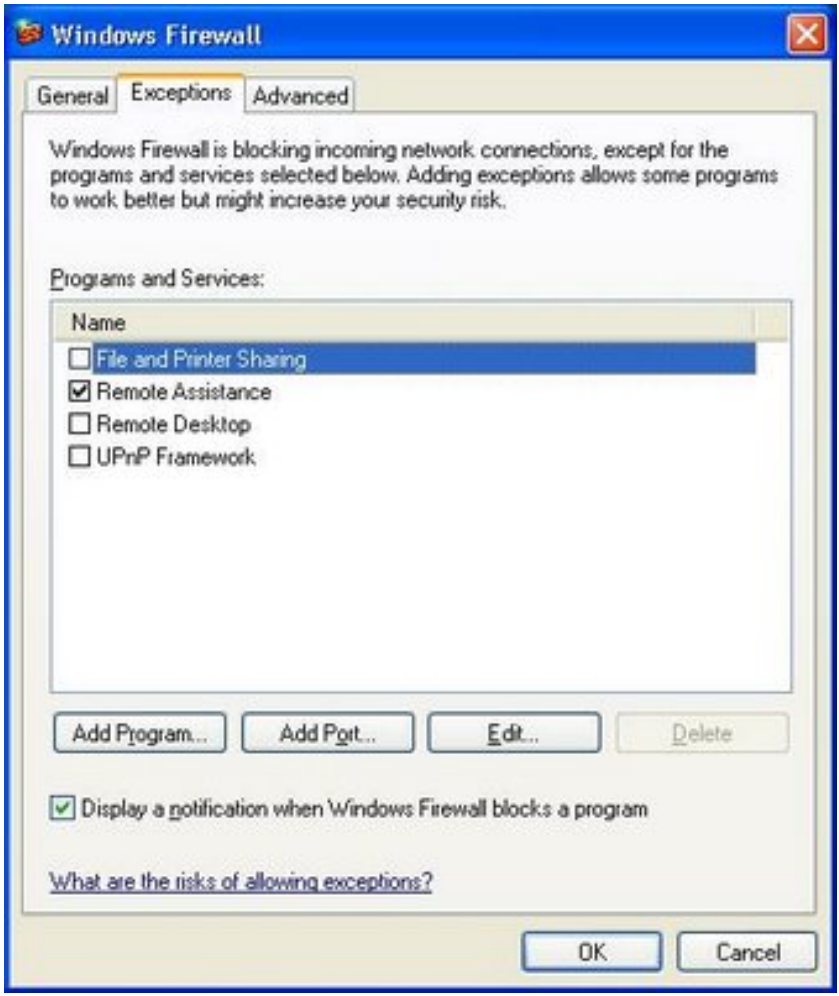
MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please set the Windows options.
Install As Windows Service
This is the recommended way to run the MySQL server on Windows.
Service Name:
MySQL
Launch the MySQL Server automatically
Include Bin Directory in Windows PATH
Check this option to include the directory containing the server / client executables in the Windows PATH variable so they can be called from the command line.
< Back
Next >
Cancel

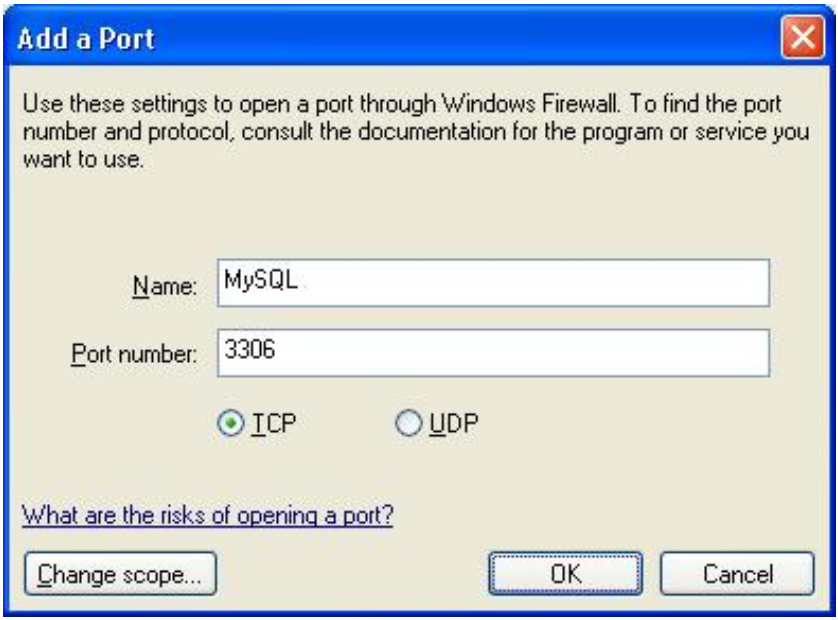
MySQL Server Instance Configuration Wizard
MySQL Server Instance Configuration
Configure the MySQL Server 5.1 server instance.
Please set the security options.
Modify Security Settings
root
New root password:
******
Enter the root password.
Confirm:
******
Retype the password.
Enable root access from remote machines
Create An Anonymous Account
This option will create an anonymous account on this server. Please note that this can lead to an insecure system.
< Back
Next >
Cancel



fire wall ஆனது enable ஆகி இருந்தால் பின்வரும் பிழை ஏற்படும்.

இதில் Port 3306-க்கு அனுமதி தரவும். இப்போது MySQL-ன் பிழை நீங்கிவிடும்.


Windows Firewall
General
Exceptions
Advanced
Windows Firewall is blocking incoming network connections, except for the programs and services selected below. Adding exceptions allows some programs to work better but might increase your security risk.
Programs and Services:
Name
File and Printer Sharing
Remote Assistance
Remote Desktop
UPnP Framework
Add Program...
Add Port...
Edit...
Delete
Display a notification when Windows Firewall blocks a program
What are the risks of allowing exceptions?
OK
Cancel


Add a Port
Use these settings to open a port through Windows Firewall. To find the port number and protocol, consult the documentation for the program or service you want to use.
Name:
MySQL
Port number:
3306
TCP
UDP
What are the risks of opening a port?
Change scope...
OK
Cancel

2

SQL Command line client-ஐ பயன்படுத்துதல்

MySQL-க்கு பல்வேறு வகையான clients உள்ளன. அவைகள் GUI கொண்ட desktop application, Internet வழியாக பயன்படுத்தக்கூடிய web-based applications மற்றும் shell வழியாக இயங்கும் text-only applications போன்றவை.

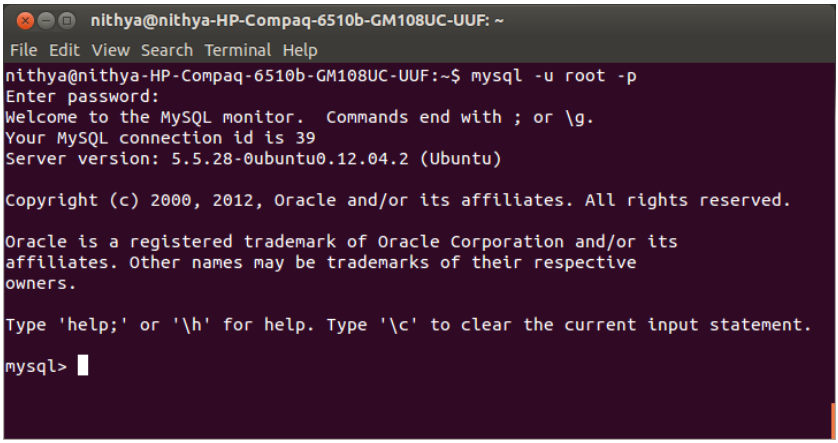
இந்தப் பகுதியில் MySQL server-வுடன் வரும் text-only command line client-ஐப் பயன்படுத்தி எவ்வாறு MySQL-ஐ அணுகுவது என்று பார்க்கலாம்.

வேறு வகையான tools பயன்பாட்டுக்கு சுலபமாக இருந்தாலும், இந்த MySQL tool-ஐ எவ்வாறு பயன்படுத்துவது என்று கற்பதன் மூலம், வேறு வகையான tool-ஐ அணுகுவதற்குத் தேவையான அடிப்படை விஷயங்களைக் கற்றுக் கொள்ளலாம்.

மேலும் இந்த MySQL tool-தான் அதன் server-ஐ பயன்படுத்துவதற்கு நம்மிடம் இருக்கும் ஒரே tool. எனவே இதனை முழுமையாக கற்றுக் கொள்வதன் மூலம் நம்முடைய வேலைத்திறனை அதிகரிக்கலாம்.

2.1 Client மூலம் Server-வுடன் இணைப்பை ஏற்படுத்தல்

MySQL-client மூலம், நமது machine-ல் இயங்கும் MySQL server-ஐ பின்வருமாறு இணைக்கலாம்.



இதற்கான முழு syntaxபின்வருமாறு.

shell> mysql [-h host] [-u user_name] [-p] [db_name]
இவ்வாறு இணைப்பை ஏற்படுத்தும்போது பயன்படுத்தப்படும் optional arguments-ஐப் பின்வருமாறு விளக்கலாம்.

- -h : எந்த host-வுடன் இணைப்பை ஏற்படுத்த வேண்டும் என்பதைக் குறிப்பிடப் பயன்படுகிறது. இதைக் குறிப்பிடவில்லையெனில், அந்த machine-ல் இருக்கும் local server-வுடன் இணைப்பை ஏற்படுத்த முயற்சிக்கும்.
- -u : Username-ஐக் குறிப்பிடப் பயன்படும். இதைக் கொடுக்கவில்லையெனில், அது தானாகவே உங்களுடைய 'UNIX' username-ஐ எடுத்துக்கொண்டு, MySQL-ல் authenticate செய்ய முயற்சிக்கும்.
- -p : இது உங்களுடைய password-ஐ குறிப்பிடப் பயன்படும். நீங்கள் உங்களுக்கான user account-ஐ உருவாக்கும்போது, password இல்லாமல் உருவாக்கியிருந்தால், இதைக் கொடுக்கத் தேவையில்லை.
- Database_name : நீங்கள் பயன்படுத்த விரும்பும் database-ன் பெயரைக் குறிப்பிட உதவும். இதை நீங்கள் கொடுக்கவில்லையெனில், வெறும் Use command-ஐப் பயன்படுத்தி நீங்கள் பயன்படுத்த

விரும்பும், database-ன் பெயரைக் குறிப்பிடலாம்.

நீங்கள், மேலும் என்னென்ன options உள்ளது என்பதை அறிய விரும்பினால்,

mysql -?

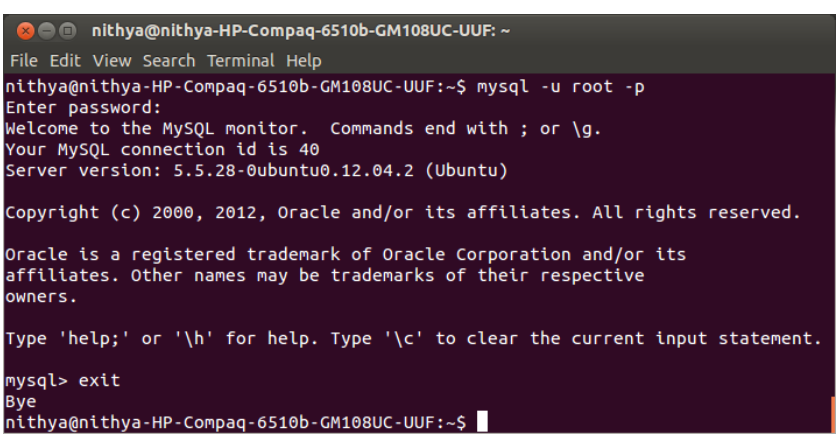
எனக்கொடுத்து shell prompt்- run செய்யவும். இது அனைத்து வகையான arguments-ஐயும் பட்டியலிடும்.

2.2 Client மூலம் server-வுடன் ஆன இணைப்பை துண்டித்தல்

பின்வருமாறு mysql> prompt-ல்

exit

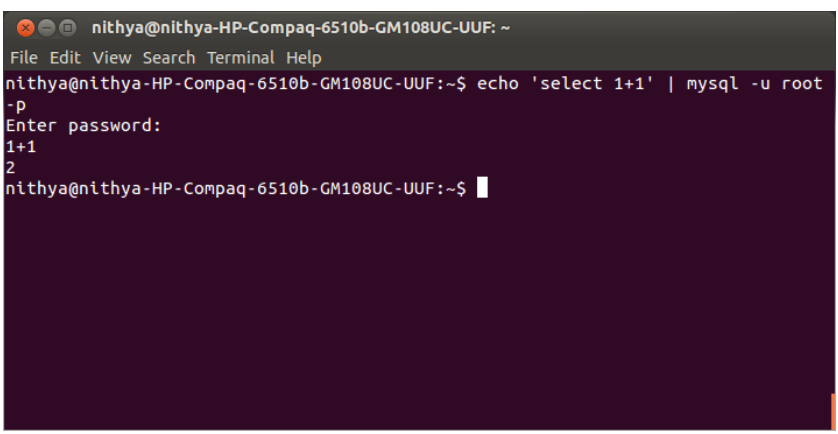
என type செய்வதன் மூலம் mysql-server-வுடனான இணைப்பை துண்டிக்கலாம்.



2.3 Batch mode-ல் client-ஐப் பயன்படுத்துதல்

ஒரு input file வழியாக queries-ஐ MySQL-க்கு அனுப்பி, அதன் result-ஐ மற்றொரு output file-க்கு அனுப்புவதைத்தான் batch mode என்கிறோம்.

பின்வரும் உதாரணத்தில், எவ்வாறு ஒரு எளிய query-ஐ MySQL-க்கு அனுப்பி அதன் result-ஐக் காண்பது எனப் பார்க்கலாம்.



இதற்கான syntaxபின்வருமாறு.

echo ‘SELECT 1 + 1’ | mysql -u some_user -p

Shell commands-ஐப் போன்றே, இந்த MySQL-ம் pipes மற்றும் input redirection-ஐ உள்ளடக்கிய பலவகையான shell

commands-ஐப் பயன்படுத்தும். பின்வரும் உதாரணத்தில் இருக்கும் shell command , ஒரு input file-லிலிருந்து அனைத்து commands-ஐயும் read செய்து அதை MySQL-க்கு அனுப்புகிறது. பின்னர் இந்த MySQL, கொடுக்கப்பட்டுள்ள username மற்றும் password-ஐ authenticate செய்தபின், பெற்றுக்கொண்ட அனைத்து commands-ஐயும் execute செய்து அதன் விடைகளை ஒரு output file-க்கு அனுப்பி வைக்கிறது.

```
mysql -u user -p < input_file > output_file
```

2.4 Client-ன் Command line-ல் இடம் பெயர்தல்

Unix,Linux மற்றும் Mac OS X போன்ற operating system-ல் இருக்கும் MySQL, GNU Readline library என்பதைப் பயன்படுத்தும். இது command line-ல் நாம் எளிதாக இடம் பெயர உதவும் ஒரு சிறந்த tool ஆகும். இந்த readline, 100-க்கும் மேற்பட்ட commands-ஐ உள்ளடக்கியது. அவற்றில் ஒருசிலவற்றைப் பின்வருமாறு காண்போம்.

- CTRL + a வரியின் தொடக்கத்துக்குச் செல்லும்
- CTRL + e வரியின் முடிவுக்குச் செல்லும்
- ALT + b ஒரு வார்த்தை பின்னோக்கிச் செல்லும்.
- ALT + f ஒரு வார்த்தை முன்னோக்கிச் செல்லும்.
- CTRL + w முன்னே உள்ள வார்த்தையை நீக்கிவிடும்.
- CTRL + u command-ஐ நீக்கிவிடும்
- CTRL + l திரையில் காணப்படும் தேவையற்ற commands-ஐ நீக்கி, திரையை சுத்தம் செய்யும்
- ALT + u கடந்த வார்த்தையை uppercase-ஆக மாற்றும்.
- ALT + l கடந்த வார்த்தையை lowercase-ஆக மாற்றும்
- ALT + _ தவறுதலாக நீக்கப்பட்ட கடைசி வார்த்தையை திரும்பக் கொண்டு வரும்.

2.5 Option file-ல் Connection Defaults- ஐ அமைத்தல்

ஒவ்வொரு முறையும் MySQL-ஐத் துவங்கும்போது, இணைப்புக்குத் தேவையான விவரங்களை திரும்ப திரும்ப கொடுப்பதற்கு பதிலாக, அந்த விவரங்களை எல்லாம் ஒரு file-ல் வைத்து, இந்த file-ஐ MySQL பயன்படுத்துமாறு செய்யலாம்.

தேவைப்பட்டால் நீங்கள் இந்த options-ஐ command line-ல் கூட மாற்றிக்கொள்ளலாம்.
ஒரு MySQL-ல் இருக்கும் அனைத்து instances-க்கும் options-ஐக் குறிப்பிடுவதற்கு my.cnf file-ஐ (இதுதான் MySQL-ன் global configuration file) மாற்றி அமைக்கவும். Unix போன்ற operating system-ல் ஒரு குறிப்பிட்ட user-க்கு இவ்வகையான options-ஐக் குறிப்பிட, அவருடைய home directory-ல் இந்த .my.cnf file-ஐ உருவாக்கி அமைக்கவும்.

எவ்வகையான file-ஆக இருந்தாலும், ஒரு file-க்குள் options-ஐ அமைப்பதற்குப் பயன்படுத்தும் முறை ஒன்றேயாகும். முதலில் அந்த form-ன் தலைப்பை மாற்றவும். பின்னர் இந்த தலைப்பை தொடர்ந்து, மற்றொரு தலைப்பு வரும்வரை அல்லது அந்த file முடியும் வரை இருக்கும் அனைத்தும், MySQL-ஐத் துவக்கும்போது அவை பயன்படுத்தக் கூடிய options ஆகும்.

இந்த file-ல் அமைக்கப்படும் options-க்கும், command line-ல் இருக்கும் options-க்கும் இடையில் மிகச்சிறிய வித்தியாசங்களே உள்ளன. அவை பின்வருமாறு.

- முன்னால் இருக்கும் 2 கோடுகள் நீக்கப்படுகின்றன. –host என்பதற்கு பதிலாக வெறும் host என்று மட்டும் எழுதப்படும்.
- ஒரு option-ன் முழு வார்த்தையும் பயன்படுத்தப்படும். அதாவது h என்பதற்கு

பதிலாக host என்று பயன்படுத்தப்படும்.
- ஒரு option-ன் பெயருக்கும் அதன் மதிப்புக்கும் இடையில் equal sign போடப்படும்.
- Comments-எல்லாம் முன்னாள் ஒரு # -ஐ வைத்து குறிப்பிடப்படும்.

மேலும் என்னென்ன உள்ளது என்பதைத் தெரிந்து கொள்ள, mysql -? என run செய்யவும்.

2.6 MySQL-ன் உதவியை நாடுதல்

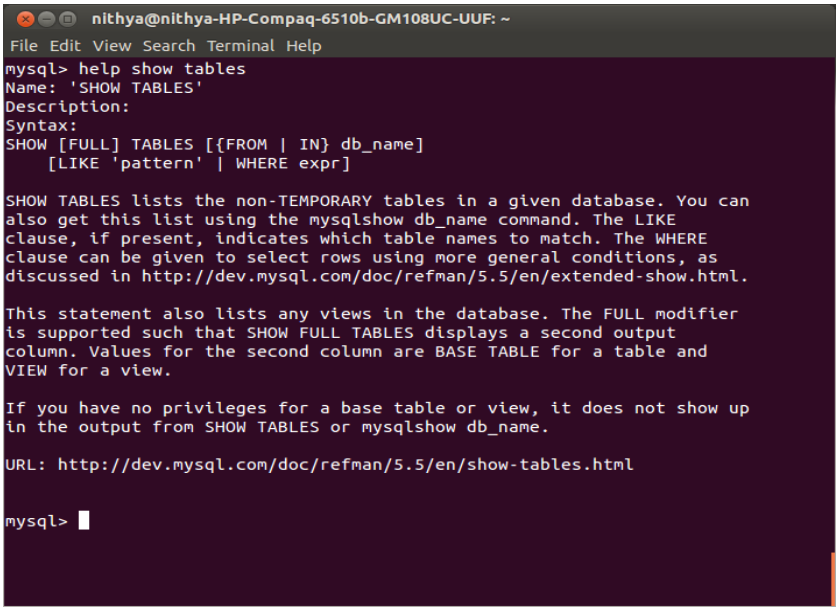
நீங்கள் MySQL command-line client-ல் வேலை பார்க்கும்போது, ஏதேனும் ஒரு command அல்லது function-ஐப் பற்றி விவரங்கள் தெரிந்துகொள்ள,

help command_or_function_name

என வைத்து enter செய்யவும். இது அந்த command அல்லது function பற்றிய வேண்டிய தகவல்களைக் கொடுக்கும். உதாரணத்துக்கு ‘show tables’ எனும் command-ன் syntax மற்றும் பயன்பாட்டினைப் பற்றி அறிய

help show tables

என வைத்து enter செய்யவும். இது பின்வரும் output-ஐக் கொடுக்கும்.



மேலும் MySQL மூலம் நமக்கு கிடைக்கும் உதவிகளின் பட்டியலைப் பார்க்க ,

help contents

என type செய்யவும். இந்த command, ஒரு புதிய MySQL server-ஐ பயன்படுத்தும்போது, நமக்கு மிகவும் பயனுள்ளதாக இருக்கும்.

2.7 Text editor உதவியுடன் commands-ஐ மாற்றுதல்

MySQL command line-ல் சிறிய queries எளிதாக enter செய்யப்பட்டாலும், பெரிய queries-ஐ அடித்து மாற்றுவது என்பது சற்று கடினமான வேலை. எனவே சற்று நீளமான queries-ஐ மாற்ற முயற்சிக்கும்போது, அந்த

query-ன் இறுதியில் \e என செய்து enter-ஐ அமுக்கவும்.

இந்த $EDITOR எனும் environment variable, நமது query-யை edit செய்யும் வகையில், editor-ல் காட்டும். பின்னர் நம் தேவைகேற்ப query-யை மாற்றி அமைத்துவிட்டு, file-ஐ save செய்தபின், editor-ஐ விட்டு வெளிவரவும்.
இந்த query-ஆனது MySQL-ல் load செய்யப்படும். ஆனால் execute செய்யப்படாது. இது execute செய்யப்படவேண்டுமெனில், semicolon (;) அல்லது \G என type செய்து enter-ஐ அமுக்கவும். இது நம்மால் edit செய்யப்பட்ட அந்த நீளமான query-யை execute செய்ய உதவும்.

2.8 Command-line-ல் Tab-ன் பயன்பாடு

ஒரு query-யை நாம் எழுதும்போது, நாம் பயன்படுத்தும் database, tables அல்லது column-ன் பெயர்களை முழுமையாகக் குறிப்பிடத் தேவையில்லை. அந்தப் பெயர்களின் தொடக்க எழுத்துக்களை மட்டும் எழுதி tab அடித்தால் போதுமானது. அந்தத் தொடக்க எழுத்துக்களை மட்டும் வைத்துக்கொண்டு, MySQL அதன் முழு பெயரையும் கண்டுபிடித்து பொருத்திவிடும்.

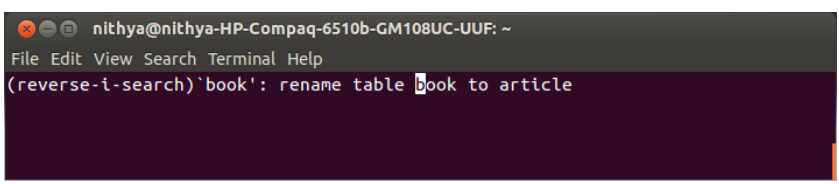
நீங்கள் ஒரு புது table-ஐ உருவாக்கினாலோ அல்லது server-ல் இருக்கும் ஆவணங்களை மாற்றினாலோ, \# என type செய்து enter-ஐ அமுக்கவும். இது அந்தப் பெயர்களின் பட்டியலை மறு உருவாக்கம் செய்யப் பயன்படும்.

2.9 Command-line history-யைப் பயன்படுத்தல்

MySQL command-line-ல் இருக்கும் மற்றுமொரு சிறப்பான அம்சம் என்னவெனில், ஒரு குறிப்பிட்ட session-ல் இதுவரை நாம் பயன்படுத்திய commands அனைத்தும் buffer பகுதியில் சேமித்து வைக்கப்படும். இந்தப் பகுதியிலிருந்து நமக்கு வேண்டிய commands-ஐ எடுத்து , அதனை நாம் மீண்டும் பயன்படுத்தலாம்.

இதுவரை நாம் என்னென்ன commands-ஐ run செய்துள்ளோம் என்பதைப் பார்க்க up arrow அல்லது ctrl + P யை அமுக்கவும். இவ்வாறு நீங்கள் பார்த்துக் கொண்டேயிருக்கும் போது, கீழ்நோக்கிச் செல்ல விரும்பினால் down arrow அல்லது ctrk + n-ஐ அமுக்கவும். தெரிகின்ற query-யை மீண்டும் run செய்ய விரும்பினால் enter-ஐ அமுக்கவும். தேவைப்பட்டால், அந்த query-யை மாற்றி அமைத்தும் கூட மீண்டும் run செய்து பார்க்கலாம்.

Command history-ல் ஏதேனும் ஒரு குறிப்பிட்ட command-ஐ தேட வேண்டுமெனில், ctrl + r என type செய்து அந்த command-ஐ முழுவதுமாக அடித்தோ அல்லது அதன் ஒரு பகுதியை மட்டும் கொடுத்தோ தேடிப் பார்க்கவும்.



நாம் கொடுத்த வார்த்தையுடன் ஒன்றுக்கும் மேற்பட்ட commands பொருந்தியதெனில், ctrl + r என மீண்டும் மீண்டும் type செய்து நாம் விரும்பிய command வரும்வரை தேடிப்பார்க்கவும்

Alt + > என்பது buffer-ல் சேமித்து வைக்கப்பட்டிருக்கும் கடைசி command-ஐ பார்க்க உதவும்.
அதேபோல் Alt + < என்பது buffer-ல் உள்ள முதல் command-ஐ பார்க்க உதவும்.

2.10 வேறுசில பயனுள்ள **Tools**

MySQL Administrator மற்றும் MySQL query browser போன்றவை நம் அனைவரது கவனத்தையும் ஈர்க்கக்கூடிய வேறுசில பயனுள்ள tools ஆகும். இவை MySQL AB-ஆல் கட்டணமின்றி இலவசமாக வழங்கப்படுகின்ற GUI Applications ஆகும். இது open source வகையைச் சேர்ந்தது. மேலும் Cross-Platform-ல் (Linux,Mac OSX &Windows) இயங்கவல்லது.

இந்த MySQL Query Browser, நாம் ad-hoc queries-ஐ செயல்படுத்திப் பார்ப்பதற்கு, ஓர் அருமையான சூழலை உருவாக்கிக் கொடுக்கும்.

MySQL Administrator-ஆனது, ஒன்று அல்லது அதற்கு மேற்பட்ட MySQL server-வுடன் தொடர்பு கொள்ளும் வகையில் ஒரு digital dashboard-ஐ அமைத்து, அதன்மூலம் productivity-யை வெகுவாக அதிகரிக்கச் செய்ய உதவும்.

3

Databases பற்றிய அறிமுகம்

Databases என்பது நமக்கு வேண்டிய data-வை எல்லாம் ஓர் ஒழுங்குபடுத்தப்பட்ட, கட்டுக்கோப்பான வடிவில் சேமிக்க உதவும் ஒரு container ஆகும். MySQL Server -இல் ஒன்று அல்லது அதற்கு மேற்பட்ட databases இருக்கும். ஒவ்வொரு database-ம் ஒரு பெயரால் குறிப்பிடப்படும். மேலும் அது ஒன்று அல்லது அதற்கு மேற்பட்ட tables -ஐக் கொண்டிருக்கும்.

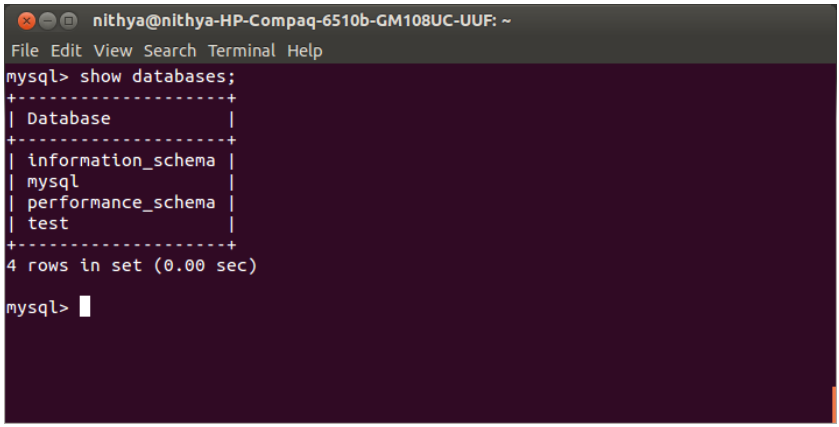
Databases -ஐ உருவாக்கி, நிர்வாகம் செய்வது என்பது மிகவும் எளிமையான வேலை. அவைகள் ஒரு சில பண்புகளுடன் கூடிய ஒரு எளிய containers ஆகவே கருதப்படுகின்றன. ஆகவே ஒரு database -ஐ எவ்வாறு உருவாக்குவது மற்றும் பயன்படுத்துவது என்பது பற்றி இங்கு கற்கலாம்.

3.1 ஒரு புதிய **database-**ஐ **server-**ல் உருவாக்குதல்

ஒரு database-ஐ உருவாக்குவதற்கு முன்பு, முதலில் நாம் server-ல் என்னென்ன databases ஏற்கனவே உள்ளன என்று தெரிந்து கொள்ள வேண்டும். அதற்காகப் பின்வரும் command பயன்படும்.

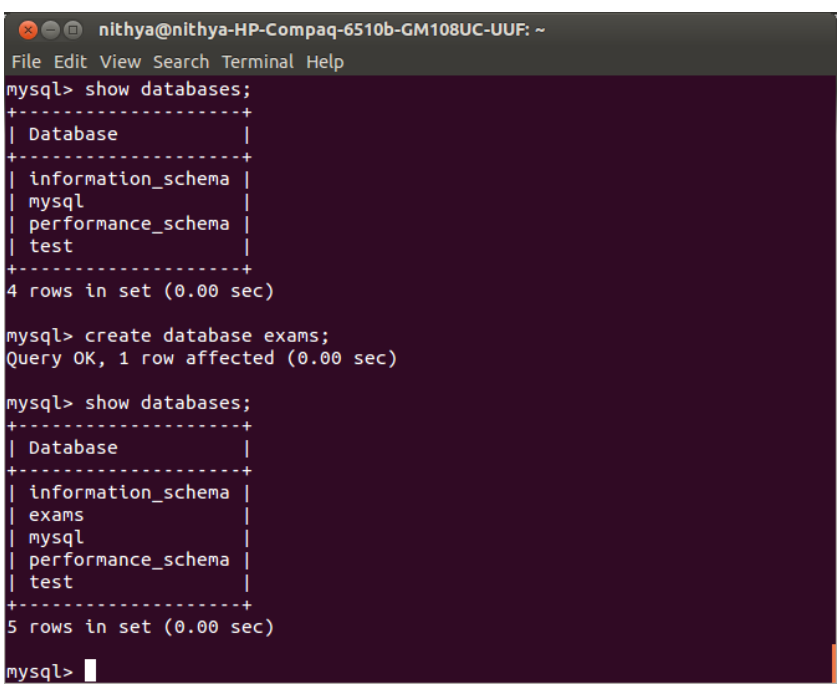
SHOW DATABASES;

இது server-ல் இருக்கும் அனைத்து databases-ன் பெயர்களையும் பட்டியலிடும். புதிதாக install செய்யப்பட்டுள்ள MySQL-ல் இந்த command பின்வரும் result -ஐக் கொடுக்கும்.



இந்தப் பட்டியலில் இல்லாத ஒரு புதிய பெயரைத்தான் நாம் உருவாக்கப்போகும் புதிய database-க்கு வைக்கவேண்டும். இதற்கான commandபின்வருமாறு.

CREATE DATABASE exams;



இந்த command, “exams” எனும் பெயர் கொண்ட ஒரு புதிய database-ஐ உருவாக்கும். ஆனால் தகவல்களை சேமிப்பதற்கு இந்த database மட்டும் போதாது. இதற்குள் tables உருவாக்கப்படவேண்டும். அப்போதுதான் இந்த database முழுமை பெரும். Tables உருவாக்குவதைப் பற்றிப் பின்னர் காணலாம்.

3.2 Database-க்கு பெயர்மாற்றம் செய்தல்

Databases பொதுவாக பெயர்மாற்றம் செய்யப்படுவதில்லை. MySQL -இல் இதற்காக தனி ஒரு SQL command -ம் இல்லை.

MySQL -இன் பழைய version -களில் பெயர்மாற்றம் என்பது பின்வரும் படிகளில் செய்யப்படுகிறது.
1. MySQL server -ஐ நிறுத்துதல்.
2. Database -ஐ குறிக்கும் directory -ஐ பெயர்மாற்றம் செய்தல்
3. Server -ஐ மீண்டும் தொடங்குதல்.

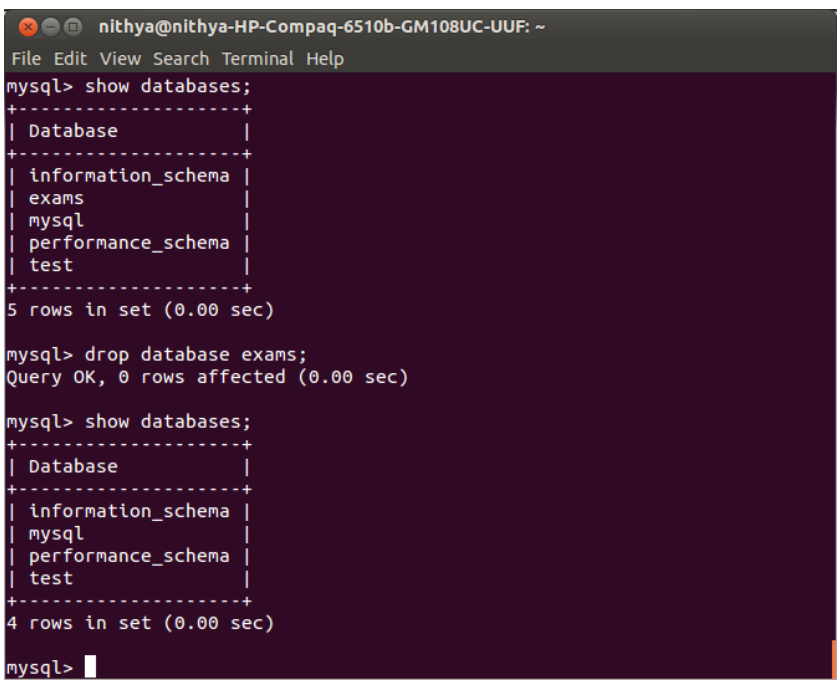
மேற்கூறிய process, MySQL -இன் தற்கால version -களில் பயன்படுத்தப்படும்போது, server ஆல் database -இல் இருக்கும் சில வகை tables -ஐ அடையாளம் கண்டுபிடிக்க இயலவில்லை. எனவே ஒரு database-க்கு பெயர்மாற்றம் செய்வது என்பது, அதனுளிருக்கும் tables-க்குப் பெயர் மாற்றம் செய்வதைப் பொறுத்து அடங்கியுள்ளது. எனவே இதைப்பற்றி பின்னர் பார்க்கலாம்.

3.3 ஒரு database-ஐ server-லிருந்து நீக்குதல்:

இந்த drop DB Command, ஒரு database -ஐ அதற்குள்

இருக்கும் அனைத்து table -களுடனும் சேர்த்து அழிப்பதற்குப் பயன்படுகிறது. இந்த command -ஐ பயன்படுத்தும் போது மிகவும் எச்சரிக்கையாக இருக்க வேண்டும். ஏனெனில் ஒரு முறை இதைப் பயன்படுத்திய பின், அழிக்கப்பட்ட tables -ஐயோ அல்லது database -ஐயோ திரும்பிக் கொண்டு வருவதற்கு வழியே இல்லை.

DROP DATABASE exams;



3.4 USE Command-ன் பயன்பாடு

MySQL install செய்யப்பட்ட பின்பு, அது பற்பல databases -ஐ உள்ளடக்கியிருக்கும். எனவே நாம் run செய்யும் query எந்த database -இல் run செய்யப்பட வேண்டும் என்பதை குறிப்பிட use command பயன்படும்.

USE library;

இதில் குறிப்பிடப்பட்டுள்ள database தானாகவே அதன் பின்வரும் query -களுக்குப் பயன்படுத்தப்படும். ஆகவே வெளிப்படையாக query-ல் database-ன் பெயரை வெளிப்படையாகக் குறிப்பிடத் தேவையில்லை. அதாவது

SELECT title FROM library.book;

எனக் குறிப்பிடத் தேவையில்லை. வெறும்

SELECT title FROM book;

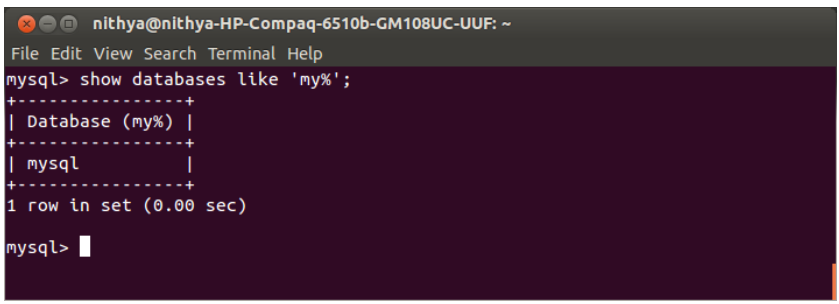
எனக் குறித்தால் போதுமானது.

3.5 LIKE Operator -ன் பயன்பாடு

ஒரு word-ல் உள்ள ஒரு பகுதியோ அல்லது அந்த word முழுவதுமாகவோ பொருந்தும் databases-ன் பெயர்களைப் பட்டியலிட இந்த Like operator -ஐப் பயன்படுத்தலாம்.

SHOW DATABASES LIKE ‘my%’;

இந்த command -ஐ நீங்கள் புதிதாக install செய்யப்பட்டுள்ள database -இல் run செய்தால் அது பின்வரும் result -ஐக் கொடுக்கும்.



இதில் Like Operator-க்கு அடுத்து நாம் குறிப்பிடும் சொல் ஒரு சாதாரண சொல். இந்த Percent(%) அல்லது Underscore(_) குறிகள் தான் இந்த சொல்லுக்கு ஒரு அர்த்தத்தைக் கொடுக்கப் போகிறது.

ஒரு சொல்லுடன் Percentage(%) -ஐ சேர்க்கும்போது, இந்த Like Command அந்த Percentage -ஐ 0 அல்லது அதன் தொடர்ச்சியாக எத்தனை எழுத்துக்கள் வேண்டுமானாலும் இணைத்து அது போன்ற சொல்லை தேட ஆரம்பிக்கும்.

ஒரு சொல்லுடன் underscore(_) -ஐ சேர்க்கும்போது, இந்த Like Command அந்த Underscore-ஐ ஏதேனும் ஒரே ஒரு எழுத்தால் மட்டுமே replace செய்து, அது போன்ற சொல்லை தேட ஆரம்பிக்கும்.

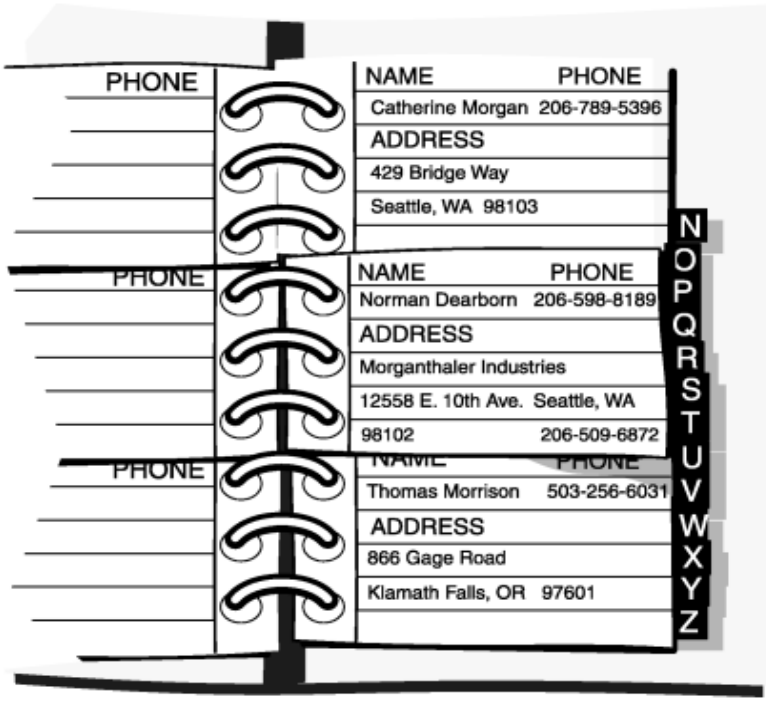
4

Tables பற்றிய அறிமுகம்

ஒவ்வொரு database-லிலும் ஒன்று அல்லது அதற்கு மேற்பட்ட tables காணப்படும். ஒவ்வொரு table-ம் ஒரு பெயரால் குறிப்பிடப்படுவதோடு அல்லாமால் அதற்கென்று ஒரு table definition -ஐயும் கொண்டிருக்கும். மேலும் இது rows மற்றும் columns எனும் format-ல் data-வை சேமிக்கும். முதலில் இந்த table எவ்வாறு உருவானது என்பது பற்றி பின்வருமாறு காணலாம்.

4.1 Table உருவான கதை

நாம் நமக்கு முக்கியமான ஒரு சிலருடைய தகவல்களை ஒரு சிறிய Address புத்தகத்தில் சேமித்து வைத்துக் கொள்ளும் பழக்கம் எல்லோருக்கும் இருக்கும். இந்த Address புத்தகமும், ஒருவருடைய பெயர், தொலைபேசி எண் மற்றும் முகவரியை எழுதி வைத்துக் கொள்ளும் வகையில் அச்சிடப்பட்டிருக்கும். மேலும் அந்தப் புத்தகத்தின் ஒவ்வொரு பக்கத்திலும் இருக்கும் index tab, அந்தப் பக்கத்தில் எந்த எழுத்தில் தொடங்கும் பெயர்கள் சேமித்து வைக்கப்பட்டுள்ளன என்பதை விளக்கும் வகையில் அமையும். உதாரணத்திற்கு அது போன்ற புத்தகத்தின் பக்கங்கள் பின்வருமாறு இருக்கும்.



இவ்வகையான தகவல்களே ஒரு table -ஐ உருவாக்குவதற்கு அடிப்படையாக அமைந்தது. அந்தப் புத்தகத்தில் இருக்கும் அச்சிடப்பட்ட விஷயங்களை (நாம் பூர்த்தி செய்ய வேண்டிய விவரங்கள்) columns ஆகவும், நாம் எழுதி வைக்கும் விஷயங்களை (நாம் பூர்த்தி செய்த விவரங்கள்) rows ஆகவும் மாற்றி ஒரு table -ஐ பின்வருமாறு உருவாக்கலாம்.

அதேபோல் அந்தப் புத்தகத்தில் இருக்கும் index-க்கு ஈடாக, இந்த table -லிலும் ஒரு column-க்கு index ஐ அமைத்து அதிலுள்ள மதிப்புகளை வரிசைப்படுத்தி வைத்துக் கொள்ளலாம். இது பின்வருமாறு அமையும்.

MySQL Table- ன் பகுதிகள் பின்வருமாறு**:**

- Table என்பது ஒருங்கிணைக்கப்பட்ட data-வின் தொகுப்புகள் ஆகும்.
- Table Body – இது row வரிசையில் data-வை உள்ளடக்கியது.
- Table definition – இது அத்தகைய row-வில் எவ்வகையான data சேமித்து வைக்கப்பட்டுள்ளது என்பதை விவரிக்கும் Columns-ஐ உள்ளடக்கியது.
- ஒவ்வொரு Column -ம் ஒரு Name(eg: book_id) மற்றும் Type(eg: Integer)-ஐ கொண்டிருக்கும். ஒவ்வொரு Type-ம் ஒரு வரையறுக்கப்பட்ட மதிப்புகளை உள்ளடக்கியது. கொடுக்கப்படும் data இந்த வரையறுக்கப்பட்ட எல்லைக்குள்தான் இருக்க வேண்டும். உதாரணத்திற்கு ஒரு Column, Integer வகையைச் சேர்ந்தது என வரையறுக்கப்பட்ட பின், அதன் மதிப்புகளெல்லாம் 1 அல்லது 6502 என்று இருக்க வேண்டுமே தவிர 3.14 என்பது போன்ற தசம எண்களாக இருக்கக் கூடாது.
- ஒவ்வொரு row-ம் தலைப்பில் உள்ள ஒவ்வொரு Column-ன் கீழ் ஒரு மதிப்பினைக் கொண்டிருக்கும். அந்த மதிப்புகள் அந்த Column-ல் வரையறுக்கப்பட்ட வகையைச் சேர்ந்தவையாக இருக்க வேண்டும். Book_id எனும் Column, SMALLINT எனும் Type-ஐ சேர்ந்தது எனில், இவ்வகையான data மட்டுமே அந்த row-வில் சேமித்து வைக்கப்பட வேண்டும்.
- வேறு சில tables-வுடன் ஒரு Relation-ல் ஈடுபட்டுள்ள Columns-ன் எண்ணிக்கை அந்த table-ன் degree எனப்படும்.
- வேறு சில tables-வுடன் ஒரு Relation-ல் ஈடுபட்டுள்ள rows-ன் எண்ணிக்கை அந்த table-ன் cardinality எனப்படும்.

Table -ஐ உருவாக்கி, நிர்வாகம் செய்வது என்பது database -ஐ உருவாக்கி நிர்வாகம் செய்வதைவிட கொஞ்சம் சிரமமான வேலை. Table -ஐ உருவாக்கும்போது, ஒரு சில சிரமமான வேலைகளைத் திட்டமிட (E.g: ஒருcolumn -

இக்கு default collation அமைத்தல், ஒரு table -இல் எவ்வளவு terabytes of data சேமித்து வைக்கலாம் என்பதை குறிப்பிடல் போன்றவை) நூற்றுக்கணக்கான options இருக்கின்றன.

எனவே நாம் Tables -ஐ உருவாக்கி, அதைப் பயன்படுத்துவதற்கான ஒரு சில அடிப்படை phrases -ஐப் பற்றி இந்தப் பகுதியில் காணலாம்.

4.2 ஒரு database-ல் table-ஐ உருவாக்குதல்

1. முதலில் நாம் table-ஐ உருவாக்குவதற்கு முன்பு, எந்த database-ல் tables உருவாக்கப்படவேண்டும் என்பதைக் குறிப்பிட வேண்டும். இதற்காகப் பின்வரும் command பயன்படுத்தப்படும்.

USE exams;

இந்த USE command, ‘exams’ எனும் database-ஐ பயன்படுத்துமாறு பணிக்கும். எனவே அடுத்தடுத்து வரும் queries அனைத்தும் இந்த database-ல் execute செய்யப்படும்.

2. ஒரு database-ல் என்னென்ன tables ஏற்கனவே உள்ளன என்று தெரிந்து கொள்ள பின்வரும் command-ஐ execute செய்யவும்.

SHOW TABLES;

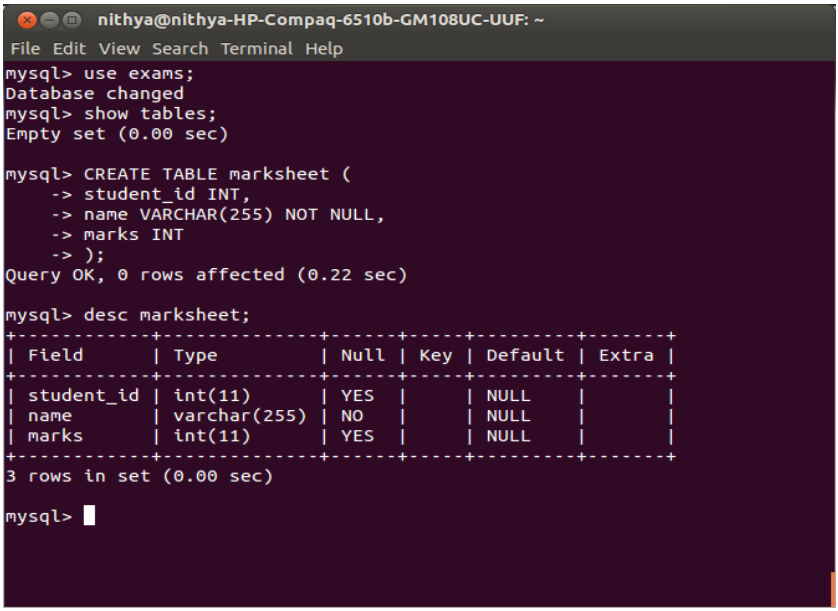
இது database-ல் இருக்கும் அனைத்து tables-ன் பெயர்களையும் பட்டியலிடும். இந்தப் பட்டியலில் இல்லாத ஒரு புதிய பெயரைத்தான் நாம்

உருவாக்கப்போகும் புதிய table-க்கு வைக்கவேண்டும்.

3. ஒரு tables -ஐ உருவாக்குவது சற்று கடினமான விஷயம் என்று **2** காரணங்களுக்காக சொல்லலாம். முதலில் tables -ஐ உருவாக்குவதற்கான syntax சற்று கடினமாகவும் மற்றும் பல elements-ஐ கொண்டதாகவும் இருக்கும். பிறகு எவ்வாறு tables -ஐ உருவாக்க வேண்டும் என்பதை விளக்கும் process அதனினும் கடினமாக இருக்கும். எனவே இந்த அனைத்து விஷயங்களையும் பயன்படுத்தி ஒரு table -ஐ உருவாக்குவதற்கான code பின்வருமாறு.

```
CREATE TABLE marksheet (
student_id INT,
name VARCHAR(255) NOT NULL,
marks INT
);
```

இந்த code -ஐப் பற்றி விளக்கமாக பின்னர் காணலாம்.



4. Table-ஐ உருவாக்கியவுடன், அதன் அமைப்பினை மீண்டும் சரிபார்க்க, பின்வரும் command-ஐ run செய்யவும்.
DESC marksheet;

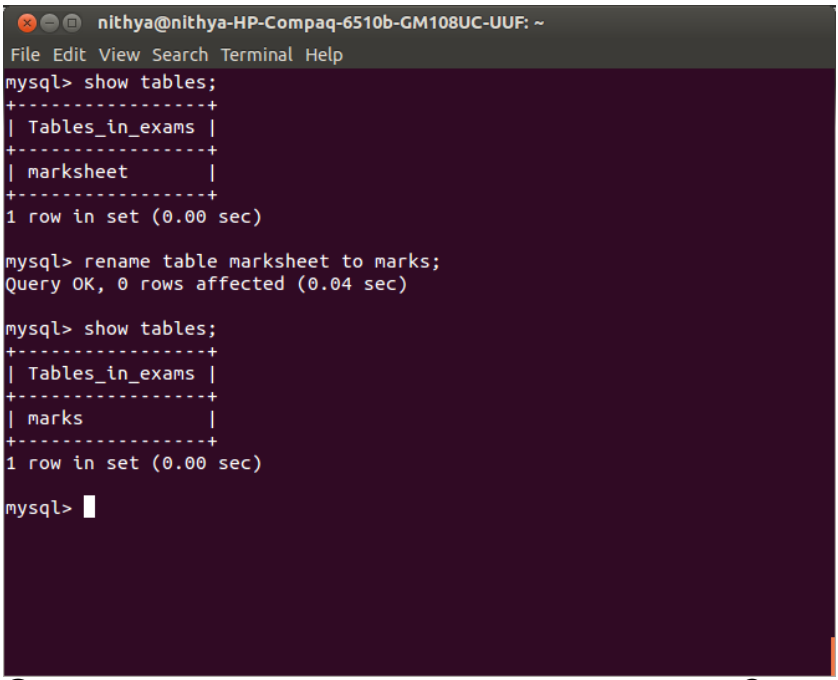
DESCRIBE marksheet;
இது நாம் உருவாக்கிய table-ன் அமைப்பினை தெளிவாகக் காண்பிக்கும்.

4.3 ஒரு database-ல் இருக்கும் table-க்கு பெயர் மாற்றம் செய்தல்

‘marksheet’ எனும் table-க்கு ‘marks’என்று பெயர் மாற்றம் செய்ய விரும்பினால், அதற்காக rename table எனும்

command-ஐ பயன்படுத்தலாம்.

RENAME TABLE marksheet TO marks;



இங்கு marksheetஎனும் table-ஆனது marks என்று பெயர் மாற்றம் செய்யப்பட்டிருப்பதைக் காணலாம்.

4.4 ஒரு database-ல் இருக்கும் table-ஐ Copy செய்தல்

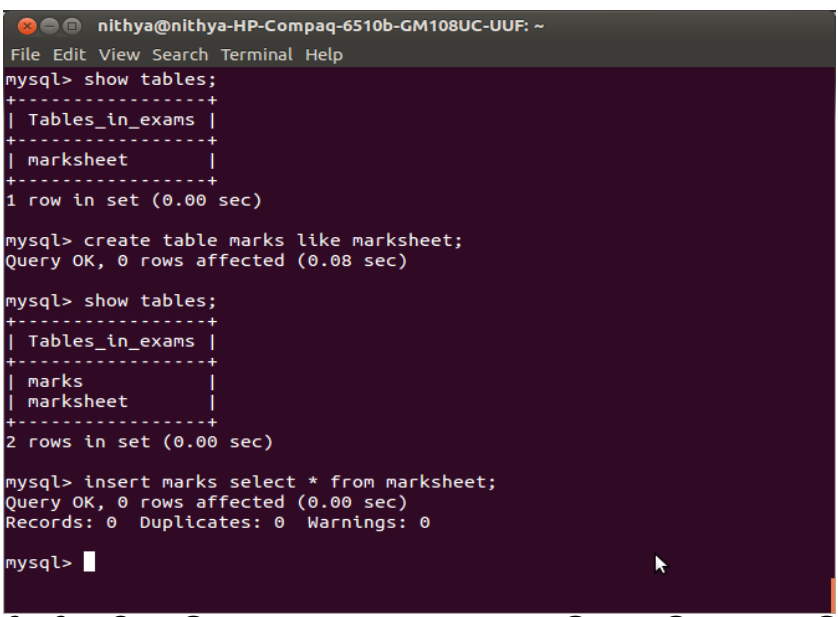
பின்னர் உள்ள create table commnad, ‘marksheet’ எனும் table-ஐப் போன்றே ‘marks’ எனும் மற்றொரு table-ஐ உருவாக்கும்.

அடுத்ததாக உள்ள insert command, ‘marksheet’ table-ல்

இருக்கும் அனைத்து தகவல்களையும் ‘marks’ table-க்குள் செலுத்த உதவும்.

CREATE TABLE new_table LIKE old_table;

INSERT new_table SELECT * FROM old_table;



மேலே கொடுக்கபட்டிருக்கும் அந்த இரண்டு queries -இம் ஒன்றாக இணைந்து, ஒரு table -இன் முழு அமைப்பையும் (அதன் indexes மற்றும் table options போன்றவைகளுடன் சேர்த்து) copy செய்கிறது.

4.5 Tables-இன் துணையுடன் ஒருdatabase-க்கு பெயர்மாற்றம்

செய்தல்

MySQL -இன் புதிய version -களில், ஒரு database-க்கு பெயர்மாற்றம் செய்வது என்பது, அதனுளிருக்கும் tables-க்குப் பெயர் மாற்றம் செய்வதைப் பொறுத்து அடங்கியுள்ளது. எனவே database-க்கு rename செய்ய பின்வரும் process பயன்படுத்தப்படுகிறது.
1. முதலில் database -இக்குள் இருக்கும் எந்த ஒரு table -இம் பயன்படுத்தப்படவில்லை என்பதை உறுதி செய்ய வேண்டும். இதற்காக MySQL Administrator tool -ஐப் பயன்படுத்தி database -ஐப் பயன்படுத்துவதற்கான அனைத்து உரிமைகளையும் நீக்கிவிட வேண்டும் இதை எவ்வாறு செய்வது என்று பின்னர் காணலாம்.

2. பின் ஒரு புதிய database -யை உருவாக்கி, பழைய database-க்கு மாற்றம் செய்யப்பட வேண்டிய பெயரை இந்த புதிய database-க்கு கொடுக்க வேண்டும்.

3. பின் பழைய database -இல் இருக்கும் ஒவ்வொரு table -இக்கும் SHOW TABLES மற்றும் RENAME TABLE command -ஐப் பயன்படுத்தி, அதை புதிய database -இக்கு மாற்றம் செய்ய வேண்டும்.
4. பழைய database -இன் அனைத்து users -இக்கும், அதே வகையான permissions -ஐ புதிய database -இல் கொடுக்க வேண்டும்.
5. அனைத்தும் ஒழுங்காக இயங்குகிறதா என test செய்து பார்க்கவும்.

6. புதிய database -இல் அனைத்தும் நாம் எதிர்பார்த்தவாறு இயங்குகிறது என உறுதி செய்யப்பட்டவுடன் பழைய database -ஐ அழித்து விடவும்.

Books எனும் database -லிருந்து library எனும் database -இக்கு book,borrower & loan எனும் 3 tables -ஐ இடமாற்றம் செய்வதற்கான code பின்வருமாறு அமையும்.

– Temporarily disable permissions

CREATE DATABASE library;

RENAME TABLE books.book TO library.book;

RENAME TABLE books.borrower TO library.borrower;

RENAME TABLE books.loan TO library.loan;

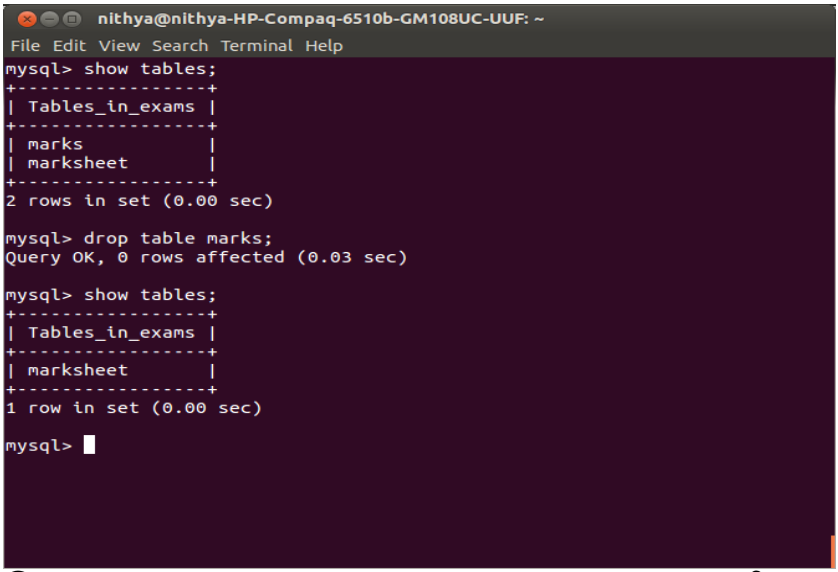
– Migrate permissions

– Re-enable permissions

4.6 ஒரு database-லிருந்து குறிப்பிட்ட table -ஐ நீக்குதல்

‘Drop table’ command மூலம் நாம் ஒரு database-லிருந்து table-ஐ நீக்கிவிடலாம்.

DROP TABLE marks;



இங்கு ‘marks’ எனும் table, ‘exams’ எனும் database-லிருந்து நீக்கப்படுகிறது.

4.7 Tables -ஐ பட்டியலிடும் விதங்கள்

பின்னர் உள்ள முதல் command, default database-ல் உள்ள அனைத்து tables -ஐயும் பட்டியலிடும்.

2வது command-ல் எந்த database-ஐப் பயன்படுத்த வேண்டும் என்பதை நாம் குறிப்பிட்டிருப்பதால், அந்த database -இல் இருக்கும் அனைத்து tables -ஐயும் பட்டியலிடும்.

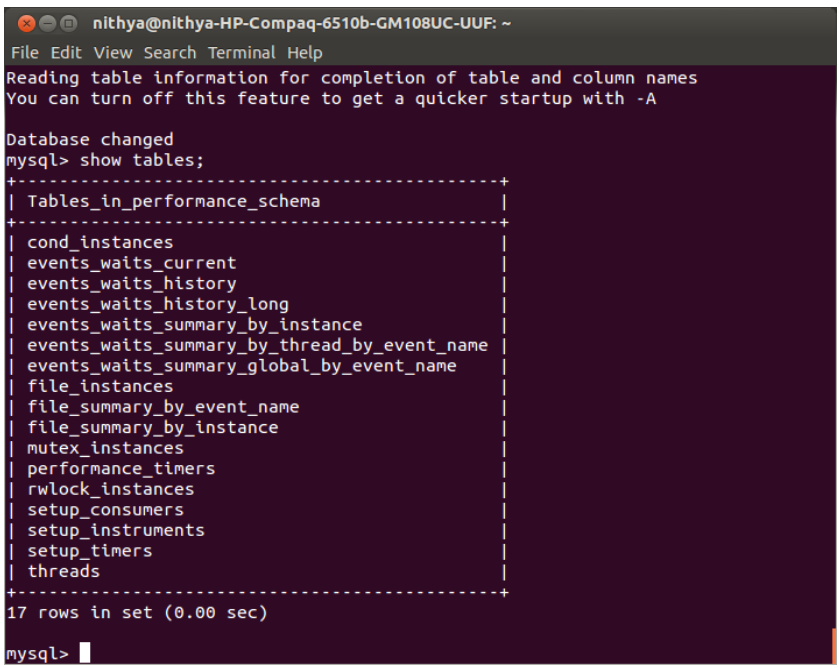
3வது command, Like operator-ல் நாம் குறிப்பிட்டிருக்கும் condition -ஐப் பொறுத்து tables-ஐப் பட்டியலிடும்.

SHOW TABLES;

SHOW TABLES IN database_name;

SHOW TABLES LIKE ‘word%’;

இந்த commands-க்கான outputs பின்வருமாறு.



5

Columns - அறிமுகம்

Table -ஐ உருவாக்கும் பொழுதே Columns -ம் உருவாக்கப்படுகின்றன. எனவே நாம் column -ல் செய்யும் ஒரு சிறிய மாற்றம் கூட table -ஐ மீண்டும் மறு உருவாக்கம் செய்வதற்கு வித்திடும். ஆகவே எப்பொழுதும் நாம் column -ஐ மாற்றம் செய்வதற்கு முன்பு, tables -ஐ ஒரு back-up எடுத்து வைத்துக்கொள்வது நல்லது.

5.1 புதியதாக ஒரு **column-**ஐ **table-**ல் சேர்த்தல்

ஏற்கனவே உள்ள table -இல் மற்றுமொரு Column-ஐ இணைப்பது என்பது மிகவும் சுலபமான வேலை. இதற்கான syntax பின்வருமாறு.

```
ALTER TABLE table_name
ADD COLUMN [column_definition];
```

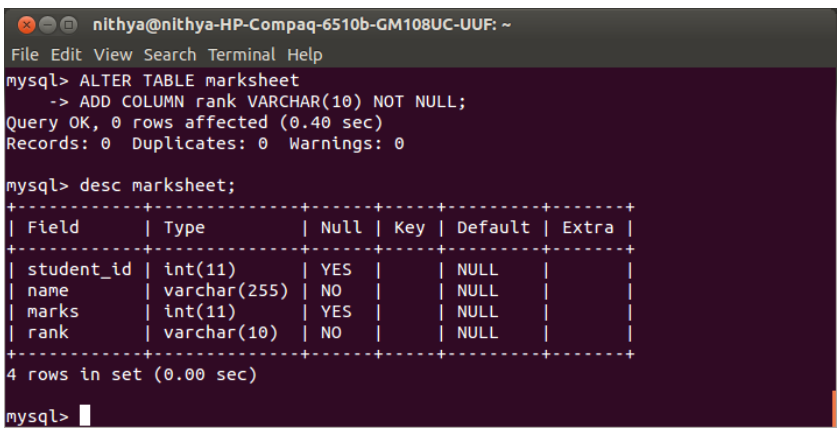
இங்கு Create table statement.-ல் நாம் கொடுக்கும் Column definition -ஐப் போலவே, இங்கும் ஒரு column definition இருப்பதைக் காணலாம். எனவே, அதன் முன் Alter table table_name என்று இணைத்தால் போதுமானது.

பின்வரும் உதாரணத்தில் எவ்வாறு rank எனும் Column,

‘marksheet’ எனும் table-வுடன் இணைக்கப்படுகிறது என்பதைக் காணலாம்.

ALTER TABLE marksheet

ADD COLUMN rank VARCHAR(10) NOT NULL;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> ALTER TABLE marksheet
-> ADD COLUMN rank VARCHAR(10) NOT NULL;
Query OK, 0 rows affected (0.40 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc marksheet;
Field	Type	Null	Key	Default	Extra
student_id	int(11)	YES		NULL	
name	varchar(255)	NO		NULL	
marks	int(11)	YES		NULL	
rank	varchar(10)	NO		NULL	
4 rows in set (0.00 sec)
mysql>

5.2 ஒரு Column-க்கு பெயர்மாற்றம் செய்தல்

ஒரு table-ல் உள்ள column-ஐ மாற்றுவது என்பது, ஒரு புதிய column-ஐ இணைப்பதற்கு நிகரான process ஆகும். ஆனால் இவ்விரண்டுக்கும் உள்ள ஒரே ஒரு வித்தியாசம் என்னவெனில், Alter table add column என்பதற்கு பதிலாக Alter table change column என்று கொடுக்க வேண்டும். இதற்கான syntax பின்வருமாறு.

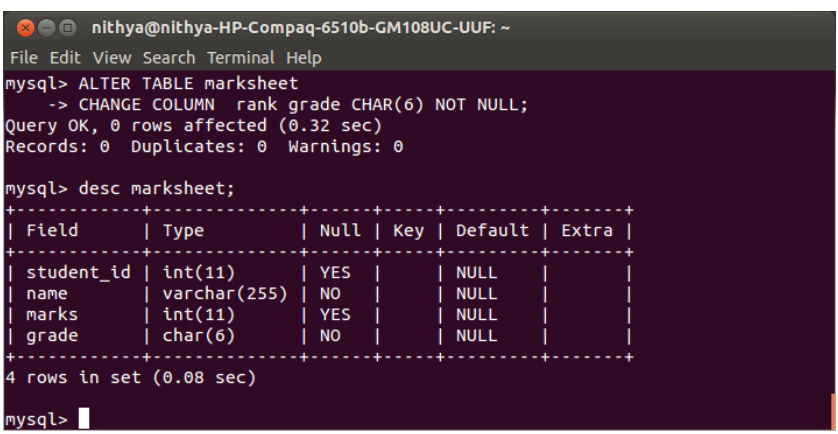
ALTER TABLE table_name

CHANGE COLUMN column_name [column_definition];

column definition என்பது அதன் type மற்றும் attributes - இக்கு முன்னர் column name -ஐக் கொண்டிருக்கும். எனவே நீங்கள் column name -ஐ அல்லாமல் column definition-ஐ மட்டும் மாற்றும் போது அந்த statement-ல், column name இரண்டு முறை காணப்படும்.

பின்வரும் query-ல், 'marksheet' எனும் table -இல் இருக்கும் 'rank' எனும் column-ன் பெயர் எவ்வாறு 'grade' என்று மாற்றப்படுகிறது எனக் காணலாம். அத்தோடுசேர்த்து, width-ம் மாற்றப்படுவதைக் காணலாம்.

```
ALTER TABLE marksheet

CHANGE COLUMN rank grade CHAR(6) NOT NULL;
```


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> ALTER TABLE marksheet
-> CHANGE COLUMN rank grade CHAR(6) NOT NULL;
Query OK, 0 rows affected (0.32 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc marksheet;
Field	Type	Null	Key	Default	Extra
student_id	int(11)	YES		NULL	
name	varchar(255)	NO		NULL	
marks	int(11)	YES		NULL	
grade	char(6)	NO		NULL	
4 rows in set (0.08 sec)
mysql>

5.3 ஒரு column-ஐ table-லிலிருந்து நீக்குதல்

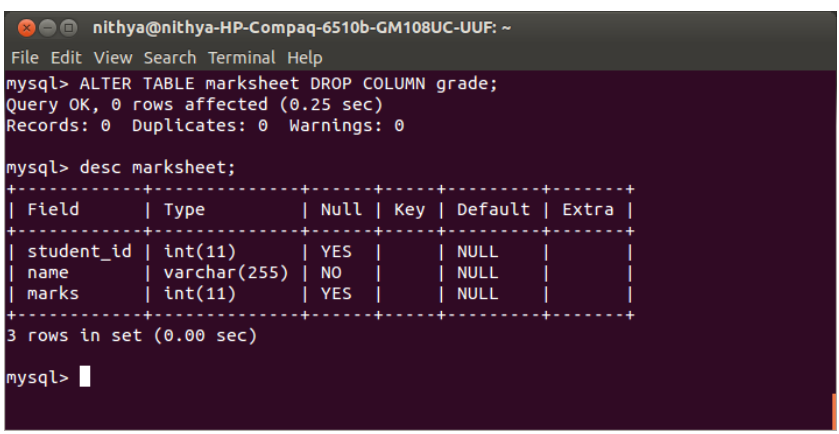
இது மிகவும் சுலபமான வேலை. இதற்கான syntax பின்வருமாறு.

ALTER TABLE table_name DROP COLUMN column_name;

ஆனால் column -ஐ ஒருமுறை நீக்கிய பின்பு, அந்த column மற்றும் அதன் row-ல் இருக்கும் அழிக்கப்பட்ட data-வை மீண்டும் கொண்டு வர முடியாது என்பதை நினைவில் கொள்ளவும்.

பின்வரும் query-ல், ‘marksheet’ எனும் table-ல் இருந்து ‘grade’ எனும் column எவ்வாறு நீக்கப்படுகிறது என்பதைக் காணலாம்.

ALTER TABLE marksheet DROP COLUMN grade;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> ALTER TABLE marksheet DROP COLUMN grade;
Query OK, 0 rows affected (0.25 sec)
Records: 0 Duplicates: 0 Warnings: 0
mysql> desc marksheet;
Field	Type	Null	Key	Default	Extra
student_id	int(11)	YES		NULL	
name	varchar(255)	NO		NULL	
marks	int(11)	YES		NULL	
3 rows in set (0.00 sec)
mysql>

6

Indexes - அறிமுகம்

Indexes என்பது table-இல் ஒரு குறிப்பிட்ட மதிப்பையோ அல்லது column -ஐயோ தேடுவதற்கு MySQL Server எடுத்துக்கொள்ளும் சிரமத்தைக் குறைக்கப் பயன்படும். மிகப்பெரிய அளவிலான tables -லிருந்து எந்த அளவுக்கு விரைவாக data -வை எடுக்க முடியும் அல்லது முடியாது என்பதை நிர்ணயிப்பது indexes ஆகும்.

6.1 ஒரு Index-ஐ table-ல் உருவாக்குதல்

Create index எனும் command -ஐப் பயன்படுத்தி ஒரு table-ல் ஏற்கனவே இருக்கும் columns-க்கு index-ஐ அமைக்கலாம். இதற்கான syntax பின்வருமாறு.

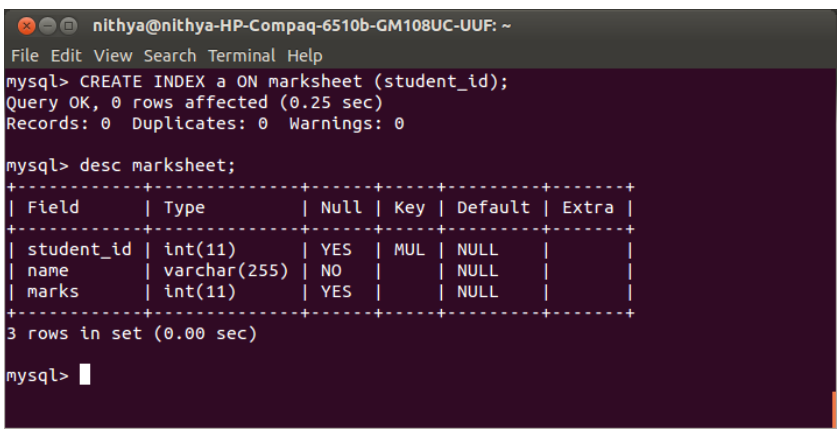
```
CREATE INDEX index_name

ON table_name (column_name, …);
```

இந்த syntax-ல், எந்த table-ல் index -ஐ உருவாக்க வேண்டும், அந்த புதிய index-ன் பெயர் மற்றும் எந்த column-ல் அதனை உருவாக்க வேண்டும் என்பது போன்ற தகவல்கள் குறிப்பிடப்பட்டிருப்பதைக் காணலாம்.

பின்வரும் query-ல் இத்தகைய தகவல்களை குறிப்பிட்டு

ஒரு index எவ்வாறு உருவாக்கப்பட்டிருக்கிறது என்பதைக் காணலாம்.

CREATE INDEX a ON marksheet (student_id);



6.2 ஒரு **Index**-க்கு பெயர்மாற்றம் செய்தல்

Index -க்கு பெயர்மாற்றம் செய்வது என்பது அரிதாக நடக்கும் ஒரு செயல் ஆகும். பொதுவாக ஒரு index அழிக்கப்படும் அல்லது அதன் definition மாற்றப்படும். எனவே ஒரு index -க்கு பெயர்மாற்றம் செய்யவேண்டுமெனில், பழைய index -ஐ அழித்து மீண்டும் புதிய பெயருடன் ஒரு index -ஐ உருவாக்குவதே சிறந்தது.

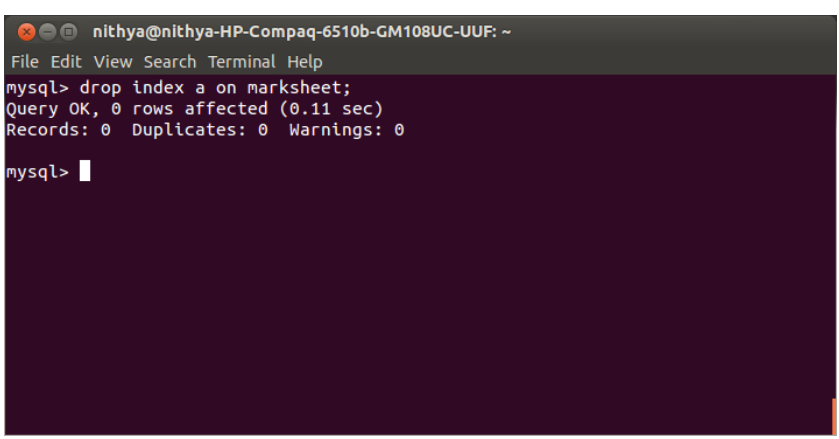
6.3 Index-ஐ அழித்தல்

Index -ஐ அழிப்பதற்கான syntax மிகவும் சுலபம். நாம் எந்த table -லிலிருந்து எந்த index -ஐ நீக்க வேண்டும் என்று குறிப்பிட்டால் போதுமானது. இதற்கான syntax பின்வருமாறு.

DROP INDEX index_name ON table_name;

பின்வரும் query-ல் எவ்வாறு ஒரு index அழிக்கப்படுகிறது என்பதைக் காணலாம்.

DROP INDEX a ON marksheet;

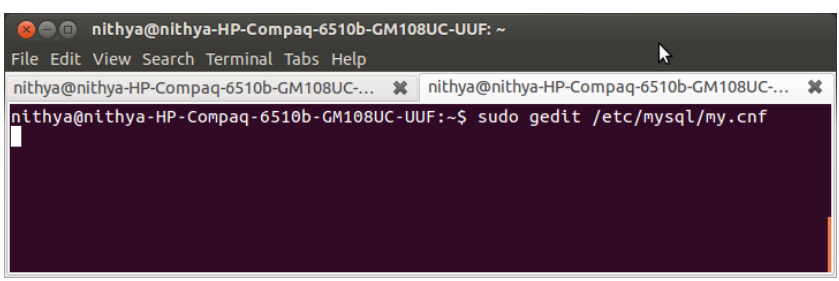


6.4 Identifiers பற்றிய விளக்கம்

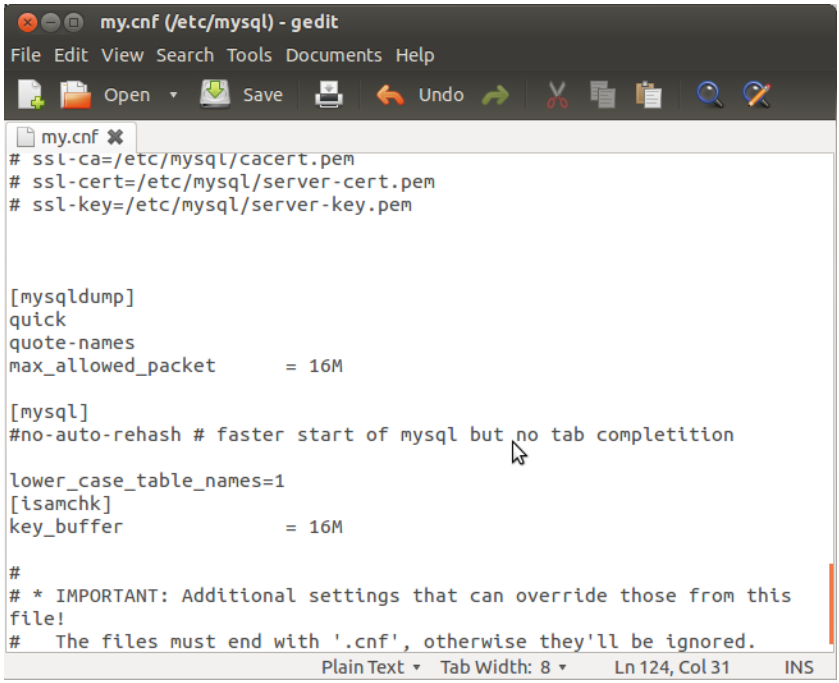
MySQL Identifiers என்பது நாம் பயன்படுத்தும் database மற்றும் column names போன்றவையே ஆகும். இவை பெரும்பாலான தருணங்களில் case sensitive ஆகும். எனவே எப்போதும் identifier-க்கு lower case letters -ஐ பயன்படுத்துவதே நல்லது. MySQL அனைத்து platform -களிலும் ஒரே மாதிரியாக நடந்துகொள்ள, my.cnf file-ல் lower_case_table_names எனும் configuration variable –ஐ 1 என அமைக்கவும்.

இதைப் பின்வருமாறு செய்யலாம்.

முதலில் shell prompt-ல் சென்று, 'my.cnf'-ஐ 'gedit'-ல் open செய்யவும்.



இது பின்வருமாறு, 'my.cnf ' file-ஐ gedit-ல் open செய்யும். அதற்குள், lower_case_table_names = 1 என type செய்யவும்.



இது அனைத்து case sensitive identifiers -ஐயும் lowercase-ஆக நடத்த உதவும்.

இந்த Identifier names -ஐப் பற்றி மேலும் விவரங்களைத் தெரிந்து கொள்ள பின்வரும் வலைதளத்தின் உதவியை நாடவும்.

http://dev.MySQL.com/doc/refman/5.0/en//identifier-case-sensitivity.html

7

Library எனும் 'மாதிரி **Databases**' -ஐ அமைத்தல்

இந்தப் பகுதியில் நாம் எவ்வாறு ஒரு database மற்றும் table-ஐ உருவாக்கி பயன்படுத்துவது என்பது பற்றி முழுவதுமாகக் கற்க உள்ளோம்.

முதலில் நாம் ஒரு database-ஐ உருவாக்கலாம். இந்த database-ஆனது ஒரு சிறிய library-க்குப் பயன்படக் கூடிய வகையில் புத்தகங்களின் விவரங்களையும் மற்றும் அந்தப் புத்தகங்கள் யாருக்கு கடனாக வழங்கப்பட்டது எனும் விவரங்களையும் கொண்டிருக்கும். எனவே இந்த database மிகவும் கடினமாக இல்லாமல், எளிதாக பின்வரும் விவரங்களை மட்டுமே கொண்டிருக்கும்.

- Library-ல் உள்ள புத்தகங்களின் தலைப்பு அதன் எழுத்தாளர் பெயர் மற்றும் அந்தப் புத்தகம் எந்த நிலையில் உள்ளது எனும் விவரம்.
- ஒரு புத்தகம் ஒரு நபருக்கு கடனாக வழங்கப்படுகிறதெனில் அந்த நபரின் பெயர், e-mail முகவரி மற்றும் எந்தத் தேதியில் அந்தப் புத்தகம் அவருக்கு கடனாக வழங்கப்பட்டது எனும் விவரம்.

7.1 Library Database-ஐ உருவாக்குதல்

Database என்பது data sets-ன் தொகுப்பு. இந்த data sets என்பது tables-ல் சேமித்து வைக்கப்படுகிறது. இந்த databases மற்றும் tables அளிக்கும் object-ஐ வைத்து MySQL-ன் பயனர்கள் அதைக் கையாளலாம்.

இந்தப் பகுதியில் நாம் library எனும் பெயர் கொண்ட ஒரு database-ஐ உருவாக்க உள்ளோம். எப்பொழுதும் ஒரு database-ஐ உருவாக்குவதற்கு முன்பு அதே பெயரில்

வேறு ஏதேனும் database உள்ளதா என்று சோதிக்க வேண்டும். இதைச் சோதிப்பதற்கு நாம் root userஆக login செய்து பின்வரும் command-ஐ அளித்தால் அது MySQL-ல் ஏற்கனவே இருக்கும் databases அனைத்தையும் பட்டியலிடும்.

SHOW DATABASES;

இப்பொழுது நாம் இந்த command-ஐ புதிதாக install செய்துள்ள MySQL-ல் run செய்திருப்பதால் அது பின்வரும் இரண்டு databases-ஐ மட்டும் பட்டியலிடும்.

- MySQL -இது administrative data மற்றும் MySQL server-ன் விவரங்களை உள்ளடக்கியது
- Test -இது புதிய ideas மற்றும் features-ஐ பாதுகாப்பாக சோதிப்பதற்கு உதவும் sandbox-ஆகப் பயன்படும்.

இந்த output-ன் மூலம் 'library' எனும் பெயரில் databases ஏதும் இல்லை எனத் தெரிந்தவுடன், இந்தப் பெயரில் புதிய database-ஐப் பின்வருமாறு உருவாக்கலாம்.

CREATE DATABASE library;

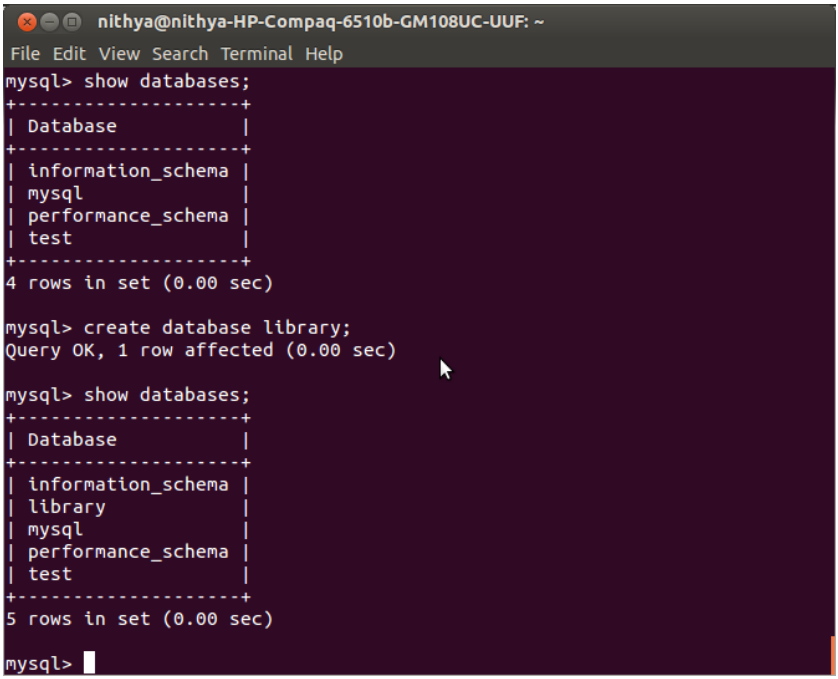
இந்த command, execute செய்யப்பட்டவுடன் MySQL client-ஆனது பின்வரும் message-ஐக் கொடுக்கும்.

Query OK, 1 row affected (0.01 sec)
இப்பொழுது நீங்கள் மீண்டும்

SHOW DATABASES;

என run செய்தீர்களானால், அது நம்முடைய library database-ஐப் பட்டியலிடுவதன்மூலம் இந்த database உருவாக்கப்பட்டு விட்டதை உறுதி செய்து

கொள்ளலாம்.



இப்போது நாம் உருவாக்கிய 'library' எனும் database, நாம் உருவாக்க இருக்கும் tablesஅனைத்துக்கும் ஒரு container போன்று செயல்படும்.

7.2 நமக்குத் தேவையான **tables**-ன் எண்ணிக்கையைத் தீர்மானித்தல்

முதலில் நாம் உருவாக்க இருக்கும் table-ல் என்னென்ன விவரங்களை சேகரிக்கப் போகிறோம் என்பதைப் பட்டியலிட வேண்டும். அப்போதுதான் நமக்கு எத்தனை tables தேவை என்பதை தீர்மானிக்க முடியும்.இது பின்வருமாறு.

- Library-ல் உள்ள புத்தகத்தின் விவரம் – அதாவது

புத்தகத்தின் தலைப்பு, எழுத்தாளர் பெயர் மற்றும் அந்தப் புத்தகத்தின் தற்போதைய நிலை.
- ஒரு புத்தகத்தை கடன் வாங்கும் நபரின் விவரம் - அதாவது அந்த நபரின் பெயர் மற்றும் முகவரி.
- கடனாக வழங்கப்பட்ட புத்தகத்தின் விவரம் – அதாவது எந்தப் புத்தகம் எந்த நபருக்கு எந்தத் தேதியில் வழங்கப்பட்டது .

எனவே இப்பொழுது நாம் table-ல் சேகரிக்க வேண்டிய முக்கியமான விவரங்களாவன:- book title, book author,book condition, person name, person email-id மற்றும் பல. இத்தகைய விவரங்கள் அனைத்தையும் நாம் ஒரே table-ல் ஒன்றாக சேகரிப்பதன் மூலம் உண்டாகும் பிரச்சனை என்னவெனில், ஒரு நபர் இரண்டுக்கும் மேற்பட்ட புத்தகங்களைக் கடனாகப் பெறும்போது , அந்த நபரின் விவரங்கள் அனைத்தும் 2 முறை வெவ்வேறு புத்தகத்திற்காக table-ல் செலுத்தப்படும். இது தேவையில்லாமல் memory space-ஐ வீணடிக்கும் வகையில் redundant data-வை table-ல் செலுத்துவதாகும்.

எனவே நமக்கு வேண்டிய எல்லாத் தகவல்களையும் ஒரே table-ல் செலுத்துவதற்கு பதிலாக, வெவ்வேறு table-ல் செலுத்தி இந்த duplicate data-ஐத் தவிர்க்கலாம்.

அதாவது புத்தகங்களின் விவரங்களும், அந்தப் புத்தகங்களைக் கடனாகப்பெற்ற நபர்களின் விவரங்களும் தான் duplicate ஆகக்கூடியவை. எனவே இவ்வாறான இரண்டு வகை விவரங்களையும் இரண்டு தனித்தனி table-ல் ஒரே ஒரு முறை செலுத்தி விட்டு, நமக்கு எப்பொழுதெல்லாம் இந்த விவரங்கள் தேவைப்படுகிறதோ அப்போதெல்லாம் இந்த table-ஐ query செய்வதன் மூலம் பெற்றுக்கொள்ளலாம்.

Reference-க்காக book மற்றும் person எனும் இரண்டு tables உருவாக்கி, பின்னர் இந்த இரண்டு table-ஐயும் refer செய்து எப்பொழுது transaction நிகழ்ந்தது என்பதை விளக்கும் வகையில் மூன்றாவதாக loan எனும் table-ஐ உருவாக்கலாம்.

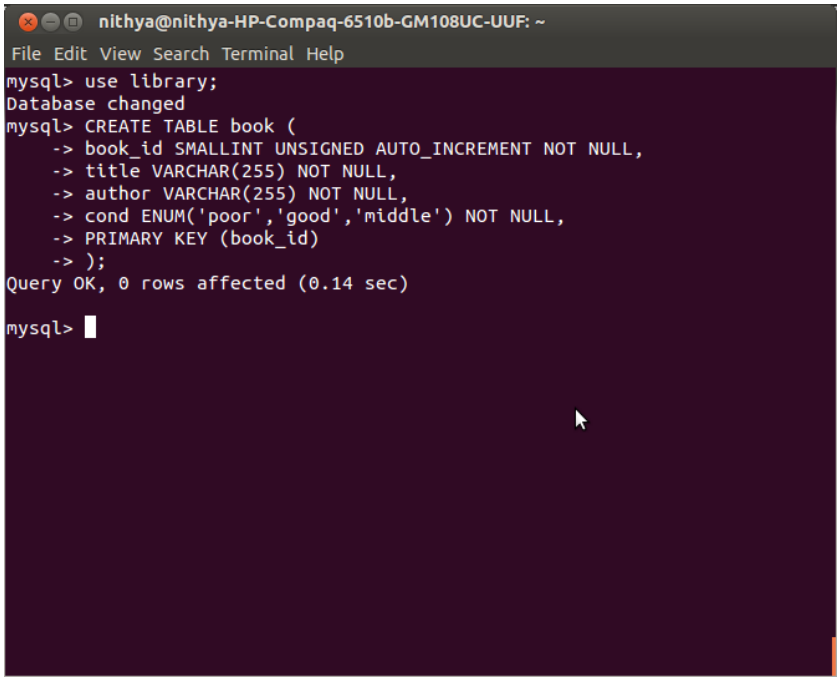
எனவே இப்பொழுது நாம் 3 தனித்தனி table-ஐ உருவாக்கப்போகிறோம்.

7.3 ‘Book’ எனும் முதலாவது table-ஐ உருவாக்குதல்

பின்வரும் command-ஆனது book எனும் table-ஐ உருவாக்கும் . இந்த query-யை command line-ல் enter செய்யும்போது, அடுத்தடுத்த line-இல் enter செய்யலாம். இது எந்த ஒரு பாதிப்பையும் ஏற்படுத்தாது.

இவ்வாறு அமைக்கப்படும் sql queries-ஆனது MySQL-ஆலும் நம்மைப் போன்ற பயனர்களாலும் எளிதில் படித்து புரிந்து கொள்ளக்கூடிய வகையில் இருக்கும் (ஆனால் string literals-க்கு இடையில் நாம் எந்த ஒரு இடைவெளியும் தரக்கூடாது. அதாவது “library” எனும் சொல்லும் “lib rary” எனும் சொல்லும் MySQL-ஐப் பொறுத்தவரை வேறு வேறு சொல் ஆகும்).

```
CREATE TABLE book (

book_id SMALLINT UNSIGNED AUTO_INCREMENT NOT NULL,

title VARCHAR(255) NOT NULL,

author VARCHAR(255) NOT NULL,

cond ENUM(‘poor’,’good’,’middle’) NOT NULL,

PRIMARY KEY (book_id)

);
```



மேலே குறிப்பிட்டுள்ள code-க்கான விளக்கம் பின்வருமாறு.

Candidate Keys:

இந்த 'book' எனும் table-ல் நாம் எல்லாப் புத்தகத்தினுடைய title, author மற்றும் condition போன்ற 3 வகையான விவரங்களைச் சேகரிக்கப் போகிறோம். எனவே இதனை 3 columns-ஆக வடிவமைக்க வேண்டும். இவை தான் இந்த table-ன் candidate keys எனப்படும்.

Primary Key:

நமது table-ல் உள்ள ஒவ்வொரு column-ன் title-ம் ஒரு candidate key எனப் புரிந்து கொண்டோம். இதில் primary

key என்பது duplicate மதிப்புகள் ஏதும் இல்லாத வகையில் அமையும் ஒரு column-ன் title ஆகும். எனவே நாம் duplicate மதிப்புகள் ஏதும் இல்லாத வகையில் இருக்கும் ஏதேனும் ஒரு candidate key-யை நமது table-ன் primary key-ஆகத் தேர்ந்தெடுக்கலாம்.

அப்படி செய்யாமல், synthetic primary key எனப்படும் வகையில் unique மதிப்புகளைக் கொண்ட ஒரு column-ஐ உருவாக்கி அதனை primary key-யாகவும் அமைக்கலாம். எனவே candidate keys என்பது நமது table-ல் இருக்கும் columns தான் என இப்பொது தெளிவாகி விட்டது. ஆகவே primary key-ம் ஒரு candidate key-ஆகவே கருதப்படும்.

இப்போது நமது “book” எனும் table-ல் பின்வரும் 3 candidate keys உள்ளன. அவை book title, book author & book condition. ஆனால் இந்த மூன்று columns-லும் duplicate மதிப்புகள் இடம்பெற வாய்ப்பு உள்ளன. Duplicate என்பது என்னவெனில் ஒரு column-ல் இருக்கும் ஏதேனும் ஒரு மதிப்பு அதே column-ல் ஒன்றுக்கும் மேற்பட்ட இடங்களில் தென்படுவதே duplicates எனப்படும். இந்த 3 column-லும் இவ்வாறான duplicate மதிப்புகள் இடம்பெற நிறையவே வாய்ப்பு உள்ளன.

ஆகவே நாம் நான்காவதாக “book_id” எனப்படும் ஒரு synthetic primary key-ஐ உருவாக்கி அதில் தொடர் வரிசையினால் அமைந்த எண்களைக் கொண்டு நிரப்பிவிடலாம். இவ்வாறு இந்த column நிரப்பப்படும்போது அதில் duplicate மதிப்புகள் இடம்பெற வாய்ப்பு இல்லை. இந்த column தான் அந்த table-ன் ஒவ்வொரு row-வையும் identify செய்ய உதவும் primary key ஆகும்.

Auto increment – primary key இன் ஓர் அங்கம்:
Auto-increment என்று அமைக்கப்படும் column -ஆனது, அந்த table-ல் ஒவ்வொரு முறை rows insert செய்யப்படும்போதும், அதற்கான வரிசை எண்களை

அந்த column-ல் தானாகவே உருவாக்கும். இந்த book எனும் table-ல் book_id எனும் primary key column-ஆனது auto increment பண்புடன் உருவாக்கப்பட்டுள்ளது.

ஒரு table-ல் ஒரே ஒரு AUTO-INCREMENT column தான் இருக்கும் மற்றும் அது primary key-ன் ஓர் அங்கமாகவும் இருக்கும் என்பதை நினைவில் கொள்க.

7.4 Data Types:

Table-ல் இருக்கும் ஒவ்வொரு column-ம் ஒரு பெயர், அந்த column-ல் சேமிக்கப் போகும் data-வின் datatype மற்றும் அந்த column-ல் அதிகபட்சம் எவ்வளவு நீளமான data சேமிக்க முடியும் என்பதைப் பொருத்து வரையறுக்கப்படுகிறது.

இந்த book எனும் table-ல் இருக்கும் நான்கு columns-ம் மேற்கூறியவற்றின் அடிப்படையில் எவ்வாறு வரையறுக்கப்பட்டுள்ளன என்பதைப் பின்வருமாறு காணலாம்

“book_id” எனும் column-ல் நாம் எண்களைச் சேமிக்கப்போகிறோம். எனவே அதன் column type-ஐ வரையறுப்பதற்கு முன்பு இந்த column-ல் அதிகபட்சம் எவ்வளவு எண்கள் சேமிக்கப்போகிறோம் என்பதை தோரயமாக கணக்கெடுக்க வேண்டும். இது library-க்கான database என்பதால் இதில் அதிகபட்சம் 50,000 புத்தகங்களைச் சேமிக்கலாம் என வைத்துக்கொள்வோம். எனவே நாம் இந்த column-க்காக UNSIGNED SMALLINT எனும் datatype-ஐப் பயன்படுத்தலாம். இது 0-லிருந்து 65,535 எண்கள் வரை சேமிக்க உதவும் ஒரு datatype ஆகும்.

அடுத்ததாக book title மற்றும் book author எனும் columns-ல் நாம் எழுத்துக்களை சேமிக்கப்போகிறோம். இவ்வாறு எழுத்துக்களைச் சேமிப்பதற்காக CHAR மற்றும் VARCHAR

எனும் இரண்டு datatypes பயன்படுத்தப்படுகின்றன. Char என்பது fixed-length மதிப்புகளை சேமிக்க உதவும் மற்றும் varchar என்பது மாறக்கூடிய மதிப்புகளை (variable length) சேமிக்க உதவும் data type ஆகும். இவை 0-லிருந்து 255 characters வரை எழுத்துக்களைச் சேமிக்க உதவும்.

நாம் condition எனும் column-லும் எழுத்துக்களை மட்டுமே சேமிக்க உள்ளோம். ஆனால் இந்த column-ல் good அல்லது poor என்பது போன்ற ஒருசில வார்த்தைகள் மட்டுமே தொடர்ச்சியாக மீண்டும் மீண்டும் இடம்பெறும். இவ்வாறு ஒரே வார்த்தைகள்தான் திரும்ப திரும்ப வெவ்வேறு row-வில் இடம்பெறும் என்று நமக்குத் தெரிந்தால் இவற்றை வரையறுப்பதற்கென்றே ENUM என்று ஒரு சிறப்பு வகை data type உள்ளது.

எனவே ஒரு column-ஐ நாம் ENUM என்று ஒரு data type கொண்டு வரையறுக்கும்போது, இந்த column-ல் என்னென்ன வார்த்தைகள் இடம்பெறலாம் என்பதையும் நாம் ஒரு list-ல் முன்னரே வரையறுத்துக் கூறிவிடவேண்டும். இவ்வாறு நாம் இந்த column-ஐ வரையறுப்பதன் மூலம் தேவையில்லாமல் நிறைய memory space வீணாவதைத் தவிர்க்கலாம்.

7.5 Column Names:

ஒரு table-ன் column-க்கு நாம் பெயர் வைக்கும்போது பின்வரும் விதிமுறைகளை மனதில் கொள்வது நல்லது.

- எப்பொழுதும் lowercase letters-ஐப் பயன்படுத்துதல்
- ஒரு table-ல் primary key-யாக இருக்கும் அதே column-தான் மற்றொரு table-ல் foreign key-யாக இருக்கும். எனவே இரண்டு table-லிலும் பயன்படுத்தப்படும் ஒரே மாதிரியான column-க்கு ஒரே மாதிரியாக பெயர்வைத்தால், primary key – foreign key

தொடர்பை நாம் சுலபமாக அடையாளம் கண்டு கொள்ள முடியும்.

- Column-ல் எவ்வகையான தகவல் சேமிக்கப்படுகிறது என்பதை, அந்த column-ன் தலைப்பை பார்த்தே புரிந்துகொள்ளும் வகையில் அந்த column-ன் title அமைக்கப்படவேண்டும். ஆனால் sql keywords-ஆக இருக்கும் date அல்லது index போன்றவற்றை நாம் column title-ஆக அமைக்கக்கூடாது.
- ஒரு column-ன் title-ஆக இரண்டு வார்த்தைகள் கொண்ட ஒரு பெயர் அமைக்கப்படுகிரதெனில், அந்த இரண்டு வார்த்தைக்கும் இடையில் Underscore(_) அமைக்க வேண்டும். இடைவெளி வரக்கூடாது.

எனவே நாம் மேற்கூறிய அனைத்து விதிமுறைகளையும் மனதில் கொண்டு “book” எனும் table-ல் இருக்கும் columns-க்குப் பின்வருமாறு பெயரிடப் போகிறோம் அவை,

- Book_id
- Title
- Author
- Cond

7.6 Person எனும் இரண்டாவது table-ஐ உருவாக்குதல்

இந்த person எனும் table-ல் நாம் ஒரு நபரின் பெயர் மற்றும் அவரின் e-mail முகவரி போன்ற விவரங்களைச் சேகரிக்கப் போகிறோம். எனவே இந்த person table-ல் person name மற்றும் email address போன்ற இரண்டு candidate keys உள்ளன. சென்ற table-ல் நாம் உருவாக்கியவாறே இந்த table-லிலும் மூன்றாவதாக person_id எனும் ஒரு primary

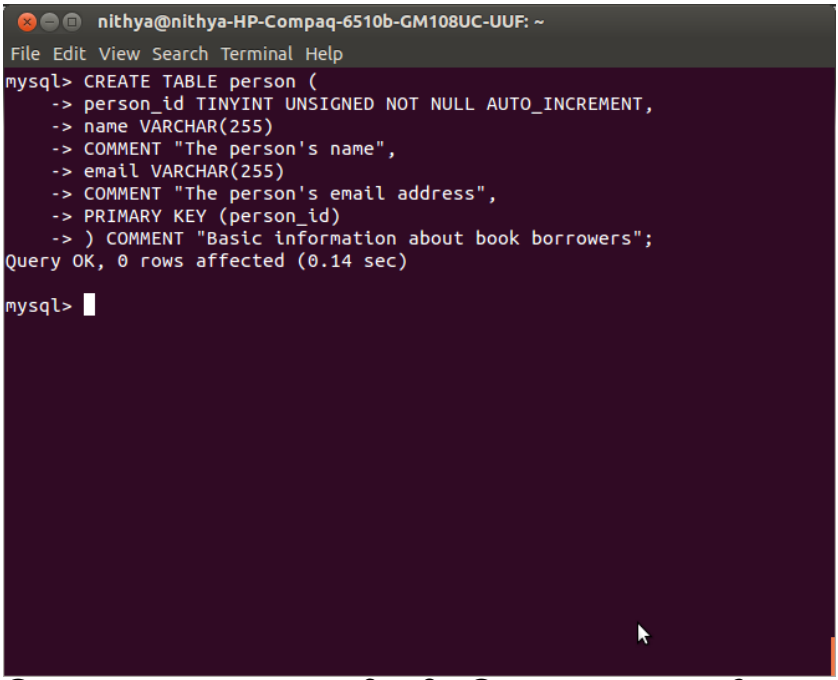
key-யை நாம் உருவாக்க வேண்டும். இது தான் இந்த table-ல் இருக்கும் அனைத்து row-வையும் refer செய்ய உதவும் primary key ஆகும்.

இந்த person_id எனும் column-ன் data type-ஐ நாம் வரையறுப்பதற்கு முன்னர், இந்த table-ல் அதிகபட்சம் நாம் எவ்வளவு data சேமிக்கப்போகிறோம் என்பதை தோராயமாக நாம் வரையறுக்க வேண்டும். இந்த library-யை பயன்படுத்தப் போகும் நபர்களின் எண்ணிக்கை எப்படியும் நூற்றுக்கணக்கில்தான் இருக்கும். எனவே இந்த column-ஐ நாம் UNSIGNED TINYINT எனும் data type கொண்டு வரையறுக்கலாம். இது 0-லிருந்து 255 எண்கள் வரை சேமிக்க உதவும் ஒரு data type ஆகும்.

அடுத்ததாக name மற்றும் e-mail போன்ற columns-ல் நாம் வெறும் எழுத்துக்களை மட்டும் சேமிக்கப்போவதால், 0-லிருந்து 255 எழுத்துக்கள் வரை சேமிக்க உதவும் VARCHAR(255) எனும் datatype-ஐயே நாம் பயன்படுத்தலாம்.

இவ்வாறாக person எனும் table-ஐ உருவாக்குவதற்கான query பின்வருமாறு அமையும்.

```
CREATE TABLE person (

person_id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,

name VARCHAR(255)

COMMENT “The person’s name”,

email VARCHAR(255)

COMMENT “The person’s email address”,

PRIMARY KEY (person_id)

) COMMENT “Basic information about book borrowers”;
```



இந்த query முன்னர் குறிப்பிட்டுள்ள query-ஐப் போன்றே இருந்தாலும் இதில் COMMENT எனும் ஒரு keyword மட்டும் புதிதாக இணைக்கப்பட்டுள்ளதைக் காணலாம். இந்த keyword, ஒரு column அல்லது table எந்த மாதிரியான data-வைத் தாங்கியுள்ளது என்பதை விவரிக்க உதவும் ஒரு optional keyword ஆகும்.

7.7 loan எனும் மூன்றாவது table-ஐ உருவாக்குதல்

இந்த table-ல் இருக்கும் candidate keys வேறுசில table-ல் இருக்கும் primary keys-ஐ refer செய்ய உதவும் foreign key-யாக அமையும். எனவே இந்த loan table-ல் அமையப்போகும் 3 candidate keys பின்வருமாறு.

- book_id எனப்படும் book table-ன் primary key
- person_id எனப்படும் person table-ன் primary key
- date_lent எனும் column, எந்தத் தேதியில் ஒரு புத்தகம் ஒரு நபருக்கு வழங்கப்படுகிறது என்பதை சேமிக்க உதவும்.

மேலும் நான்காவதாக loan_id எனும் ஒரு primary key-யை இந்த table-க்காக நாம் உருவாக்க வேண்டும். எனவே இதற்கான syntax பின்வருமாறு அமையும்.

CREATE TABLE loan (

loan_id INT UNSIGNED NOT NULL AUTO_INCREMENT,

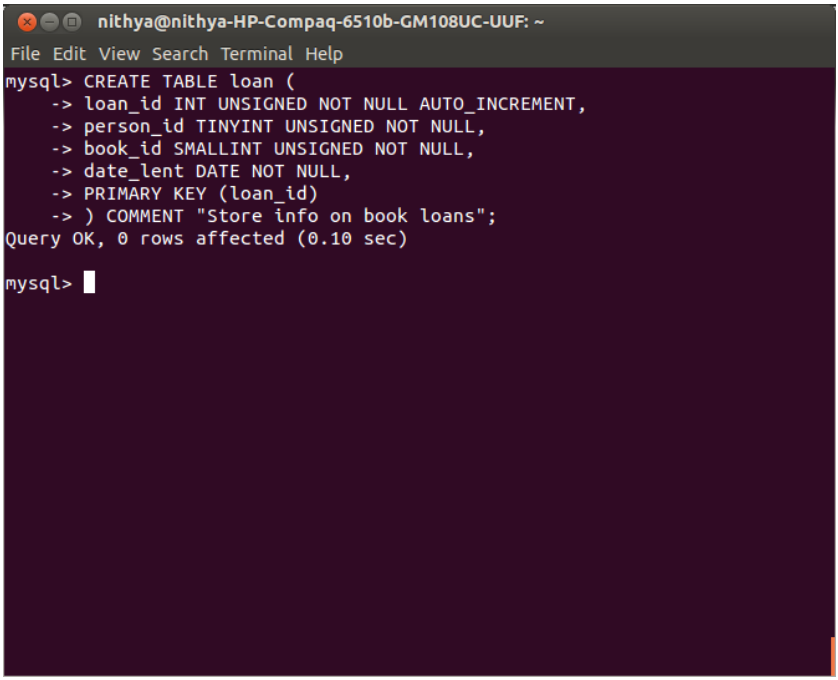
person_id TINYINT UNSIGNED NOT NULL,

book_id SMALLINT UNSIGNED NOT NULL,

date_lent DATE NOT NULL,

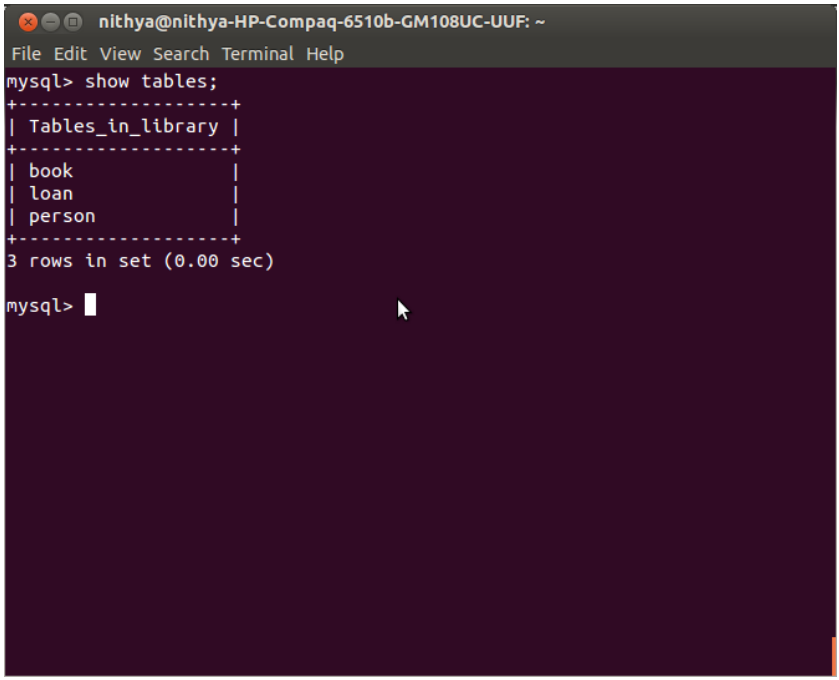
PRIMARY KEY (loan_id)

) COMMENT “Store info on book loans”;



இந்த query-ல் இருக்கும் DATE எனும் datatype தவிர மற்ற அனைத்து data type பற்றியும் நாம் முன்னரே பார்த்து விட்டோம். இந்த DATE எனும் data type-ஆல் வரையறுக்கப்படும் column-ஆனது 1000-01-01-ல் தொடங்கி 9999-12-31 வரையிலான தேதிகள் வரை சேமிக்க உதவும் data type ஆகும்.

இவ்வாறாக book, person மற்றும் loan எனும் 3 tables-ம் ‘library’ எனும் database-க்குள் உருவாக்கப்பட்டுவிட்டது.



அடுத்ததாக இந்த tables-க்குள் எவ்வாறு data-வை செலுத்துவது என்று பார்க்கலாம்.

8

Library எனும் மாதிரி **Database**-ல் **data**-வை செலுத்துதல்

ஒரு table-ல் data-வை செலுத்துவதற்கு பல்வேறு வழிமுறைகள் இருந்தாலும், பொதுவாக insert statement மூலம் செலுத்துவது நடைமுறையில் அதிகமாகப் பயன்படுத்தப்படுகிறது. இதற்கான syntax பின்வருமாறு.

INSERT [INTO] [db_name.]table_name

(list, of, columns, …)

VALUES (list, of, values, …)

- Insert [into] – இதுதான் command-ன் தொடக்கம். Into என்பது optional keyword ஆகும்.
- table_name – நாம் data-வை எந்த table-க்குள் நுழைக்கவேண்டுமோ அந்த table-ன் பெயர்.
- (list,of,columns….) – எந்தெந்த column-ல் data வை செலுத்த வேண்டுமோ அந்த columns-ஐ ஒன்றன் பின் ஒன்றாக parenthesis-க்குள் கொடுக்க வேண்டும். இதற்குள் கொடுக்கப்படாத columns ஒரு row உருவாக்கப்படும் போது default மதிப்புகளைப் பெற்றுவிடும்.
- values – இந்த command மூலம் MySQL இதன் தொடர்ச்சியாக ஒருசில மதிப்புகள் வரப்போகிறது என்று தெரிந்து கொள்ளும்.
- (list, of, values,….) – எந்தெந்த மதிப்புகள் column-ல் கொடுக்கப்பட வேண்டுமோ, அந்த மதிப்புகளை ஒன்றன் பின் ஒன்றாக ஒரு parenthesis -க்குள் கொடுக்க வேண்டும்.

இந்த insert statement ஆனது table-ன் இறுதியில் ஒரு row-

வை insert செய்யும். அதன் பின்னர் கொடுக்கப்படும் மதிப்புகளை அந்த row-வில் செலுத்திவிடும்.

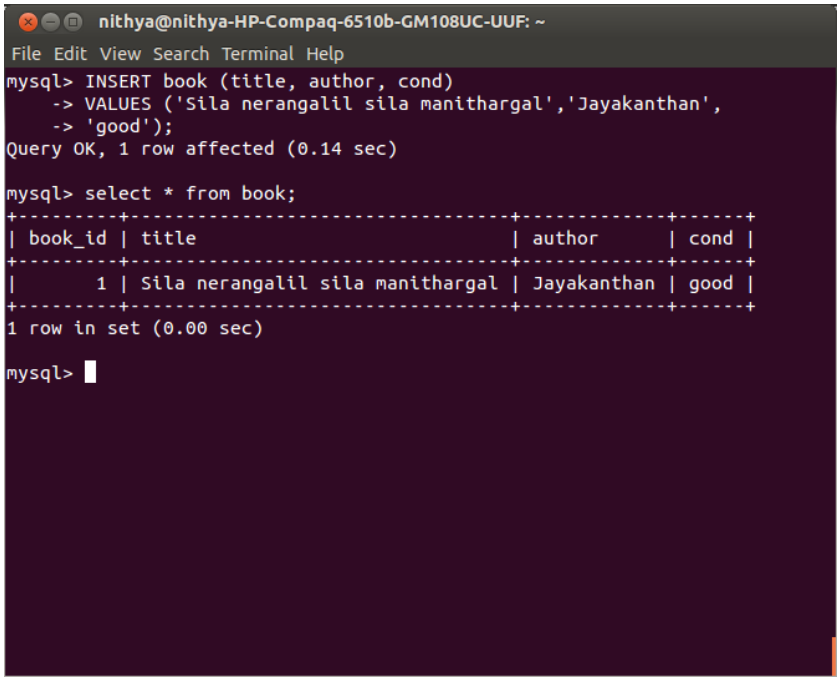
8.1 book எனும் table-ல் data-வை செலுத்துதல்

book எனும் table-ல் ஒரு row எவ்வாறு செலுத்தப்படுகிறது என்பதை பின்வரும் உதாரணத்தின் மூலம் காணலாம்.

INSERT book (title, author, cond)

VALUES (‘Sila nerangalil sila manithargal’,’Jayakanthan’,

‘good’);



நாம் செலுத்திய தகவல்கள் table-க்குள் சென்றுவிட்டதா என்பதை select statement மூலம் சரிபார்த்துக் கொள்ளலாம்.

இதில் Insert எனும் keyword-ஐத் தொடர்ந்து நேரடியாக “book” எனும் tablename கொடுக்கப்பட்டுவிட்டது. இடையில் இருக்கும் INTO எனும் optional keyword நீக்கப்பட்டிருப்பதைக் காணலாம். அடுத்ததாக book எனும் tablename-ஐத் தொடர்ந்து, எந்தெந்த column-ல் மதிப்புகள் செலுத்தப்பட வேண்டும் எனும் column list கொடுக்கப்பட்டுள்ளது. ஒரு row-ஆனது insert செய்யப்படும்போது இந்த list-க்குள் இல்லாத columns-க்கு default மதிப்புகள் insert செய்யப்படும்.

இந்த list-க்குள் “book_id” எனும் primary key column-ஐக் கொடுக்கத்தேவையில்லை. ஏனெனில் ஒவ்வொரு row, insert செய்யப்படும்போதும் இந்த column-ஆனது auto-increment மூலம் மதிப்புகளைப் பெற்றுவிடும் எனவே இந்த column-ஐ வெளிப்படையாக list-க்குள் கொடுக்கத்

தேவையில்லை.

இந்த list-ம் optional-ஆகவே பயன்படுத்தப்படுகிறது. இந்த list கொடுக்கப்படவில்லை என்றாலும், மதிப்புகள் அனைத்தும் table-ல் இருக்கும் columns-ல் தொடர்ச்சியாக செலுத்தப்பட்டுவிடும். ஆனால் எந்த மதிப்புகள் எந்த column-ல் செலுத்தப்படுகின்றன எனும் விஷயத்தில் பெரும் குழப்பம் ஏற்பட நேரிடும். எனவே இதனைத் தவிர்ப்பதற்காக இவ்வாறு வெளிப்படையாக கொடுத்து விடுவது நல்லது.

கடைசியாக VALUES எனும் keyword-ஐத் தொடர்ந்து எந்தெந்த மதிப்புகள் table-ல் செலுத்தப்பட வேண்டுமோ அந்த மதிப்புகள் அனைத்தும் ஒரு parenthesis-க்குள் கொடுக்கப்படும். இந்த parenthesis-க்குள் இருக்கும் ஒவ்வொரு மதிப்பும் ஒரு comma-ஆல் வேறுபடுத்தப்பட்டிருக்கும். மேலும் இந்த மதிப்புகள் எழுத்துக்களாக இருந்தால் அதனை quotes-க்குள் கொடுக்க வேண்டும். எண்களாக இருந்தால் அது தேவையில்லை.

அடுத்ததாக, ஒன்றுக்கும் மேற்பட்ட rows-ஐ table-ல் செலுத்துவதற்கு, மதிப்புகளைக் கொண்டுள்ள parenthesis-ஐ தொடர்ச்சியாக comma-ஆல் இணைத்துக் கொண்டே செல்லலாம். இறுதியாக உள்ள parenthesis-ன் முடிவில் semicolon(;) -ஐ அமைப்பதன் மூலம் நாம் கடைசியாக insert செய்யப்படும் row-வை சுட்டிக்காட்ட முடியும்.

பின்வரும் உதாரணத்தில் book எனும் table-ல் எவ்வாறு 3 rows செலுத்தப்படுகின்றன என்பதைக் காணலாம்.

INSERT book (author, title, cond)

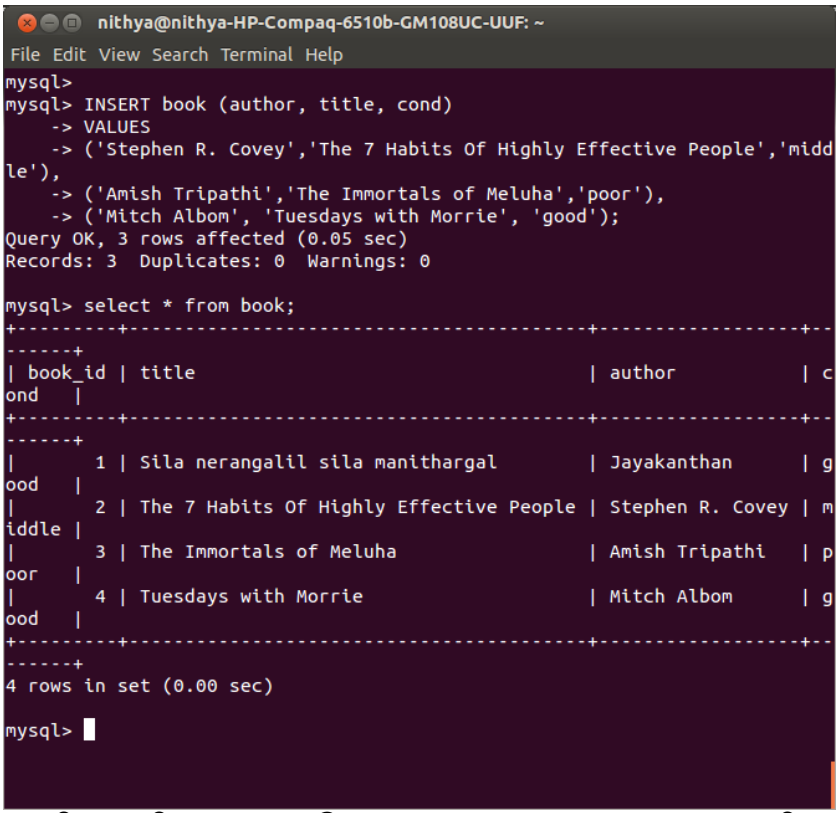
VALUES

(‘Stephen R. Covey’,’The 7 Habits Of Highly Effective

People’,’middle’),

(‘Amish Tripathi’,’The Immortals of Meluha’,’poor’),

(‘Mitch Albom’, ‘Tuesdays with Morrie’, ‘good’);


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql>
mysql> INSERT book (author, title, cond)
-> VALUES
-> ('Stephen R. Covey','The 7 Habits Of Highly Effective People','middle'),
-> ('Amish Tripathi','The Immortals of Meluha','poor'),
-> ('Mitch Albom', 'Tuesdays with Morrie', 'good');
Query OK, 3 rows affected (0.05 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from book;
book_id	title	author	cond
1	Sila nerangalil sila manithargal	Jayakanthan	good
2	The 7 Habits Of Highly Effective People	Stephen R. Covey	middle
3	The Immortals of Meluha	Amish Tripathi	poor
4	Tuesdays with Morrie	Mitch Albom	good
4 rows in set (0.00 sec)
mysql>

தனித்தனியாக ஒவ்வொரு query எழுதுவதற்கு பதிலாக மேற்கூறிய முறை மூலம் நாம் ஒரே query-ல் திறமையாக பல rows-ஐ insert செய்யலாம்.

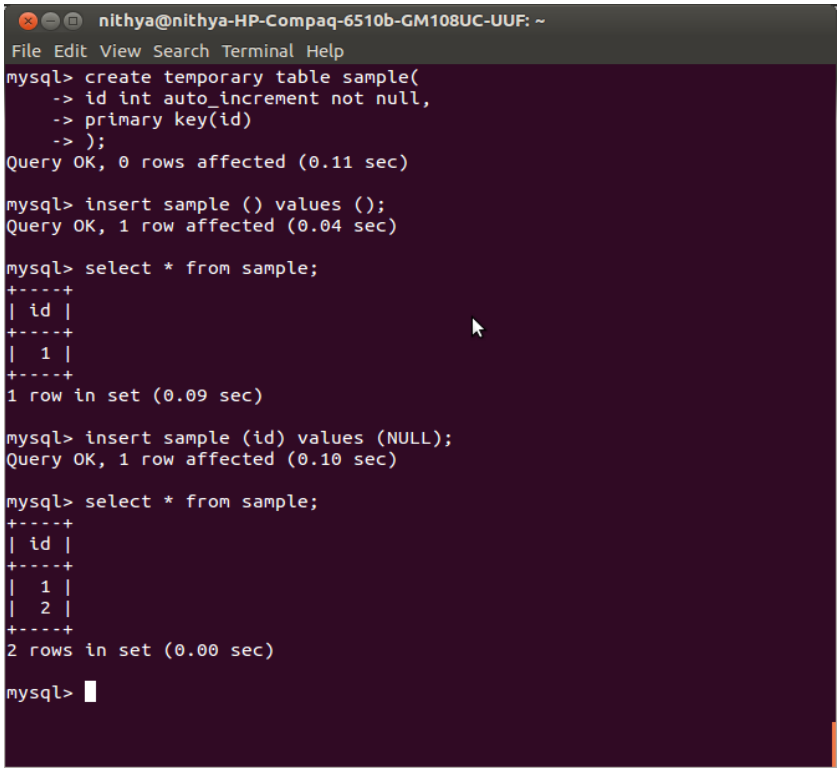
8.2 Auto increment column-ல் data-ஐ செலுத்துல்

நீங்கள் ஒரு row -வை insert செய்யும் போது, auto-increment என்று இருக்கும் column-க்கு ஏதேனும் ஒரு மதிப்பினைக்

கொடுத்தீர்களானால், அந்த column அந்த மதிப்பினைப் பெற்றுவிடும். அப்படி இல்லையெனில், அந்த column தானாகவே ஒன்றன்பின் ஒன்றாக இருக்கும் தொடர்ச்சியான எண்களால் நிரப்பப்படும். இந்த book எனும் table-ல் book_id எனும் column auto-increment மூலம் தொடர்ச்சியான எண்களால் நிரப்பப்பட்டிருப்பதைக் காணலாம்.

ஒரு row-வை நாம் delete செய்யும்போது auto-increment column-ல் இருக்கும் தொடர்ச்சியான எண்களுக்கிடையில் ஓர் இடைவெளி ஏற்படும். இந்த இடைவெளி பார்ப்பதற்கு வித்தியாசமாக இருந்தாலும், தொடர்ச்சியான எண்களை அடுத்தடுத்த rows-க்கு பயன்படுத்தாமல் இருப்பது, அந்த data-வின் நிலைப்புத்தன்மையை உறுதிபடுத்தும்.
ஒரு row -ஆனது insert செய்யப்படும்போது, auto-increment column-க்கு எந்த ஒரு மதிப்பும் கொடுக்காவிட்டாலோ அல்லது NULL எனும் மதிப்பினைக் கொடுத்தாலோ அது ஒன்றாகவே கருதும். இதைப் பின்வரும் எடுத்துக்காட்டில் காணலாம்.

```
CREATE TEMPORARY TABLE demo (

id INT NOT NULL AUTO_INCREMENT,

PRIMARY KEY (id)

);

INSERT demo () VALUES ();

SELECT id FROM demo; # id contains 1

INSERT demo (id) VALUES (NULL);

SELECT id FROM demo; # id contains 1 and 2
```



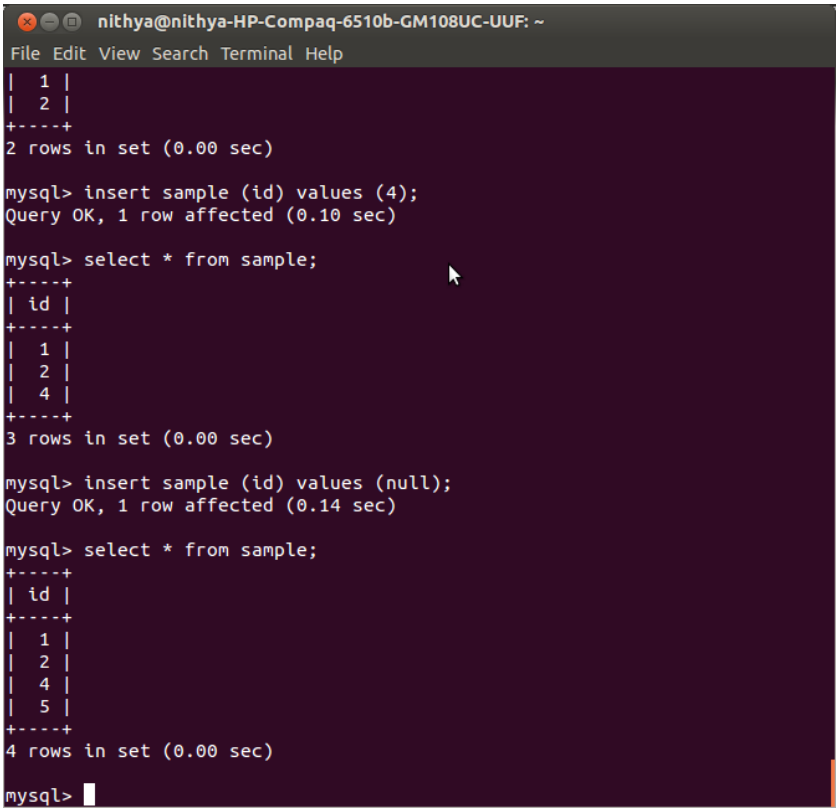
INSERT demo (id) VALUES (4);

SELECT id FROM demo; # id contains 1, 2 and 4

INSERT demo (id) VALUES (NULL);

SELECT id FROM demo; # id contains 1, 2, 4

and 5



8.3 Person எனும் table-ல் data-வை செலுத்துதல்

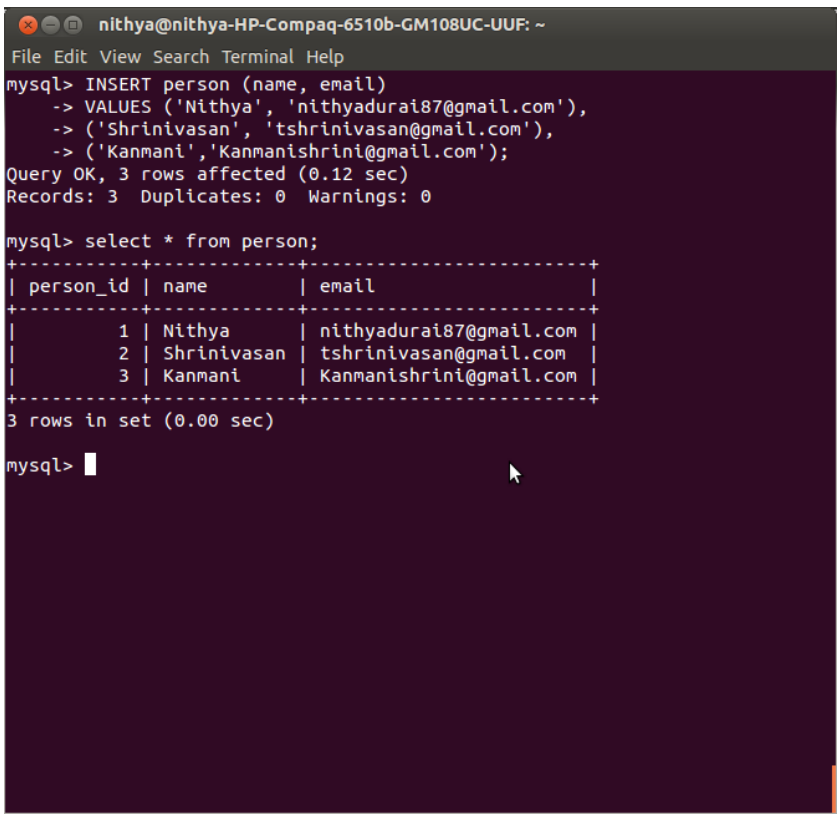
அடுத்ததாக person எனும் table-ல் புத்தகங்களைப் பெறவிரும்பும் நபர்களின் விவரங்களைப் பின்வருமாறு insert செய்யலாம்.

INSERT person (name, email)

VALUES (‘Nithya’, ‘nithyadurai87@gmail.com‘),

(‘Shrinivasan’, ‘tshrinivasan@gmail.com‘),

(‘Kanmani’,‘Kanmanishrini@gmail.com‘);


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> INSERT person (name, email)
-> VALUES ('Nithya', 'nithyadurai87@gmail.com'),
-> ('Shrinivasan', 'tshrinivasan@gmail.com'),
-> ('Kanmani','Kanmanishrini@gmail.com');
Query OK, 3 rows affected (0.12 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> select * from person;
person_id	name	email
1	Nithya	nithyadurai87@gmail.com
2	Shrinivasan	tshrinivasan@gmail.com
3	Kanmani	Kanmanishrini@gmail.com
3 rows in set (0.00 sec)
mysql>

நாம் செலுத்திய தகவல்கள் table-க்குள் சென்றுவிட்டதா என்பதை select statement மூலம் சரிபார்த்துக் கொள்ளலாம்.

8.4 ஒரு file-ல் உள்ள queries மூலம் data-வை table-ல் செலுத்துதல்

நாம் பார்த்த பெரும்பாலான SQL commands நேரடியாக MySQL command-line -ல் run செய்யப்பட்டவை. இப்போது இந்த commands-ஐ எல்லாம் ஒரு புதிய text file-ல் சேமித்து வைத்துக் கொண்டு, அதனை MySQL (அல்லது வேறு எதாவது client) மூலமாக எவ்வாறு run செய்யலாம் என்று பார்க்கலாம். இவ்வாறு செய்யப்படுவதை 'Batch mode' என்கிறோம். இதற்கான syntax பின்வருமாறு.

shell> MySQL -u username -p db_name < file_name.sql

இந்த syntax பின்வருமாறு விளக்கப்படுகிறது.

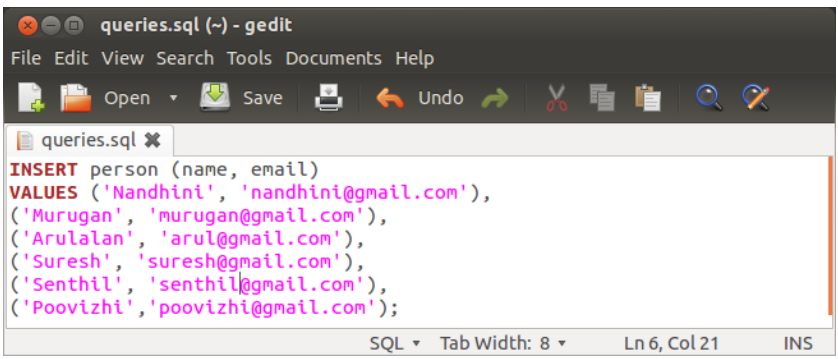
1. shell> prompt -ல் MySQL server-ஐ கொடுக்கப்பட்டுள்ள username-ஐ வைத்து connect செய்யவும்.
2. பின் அதனுடன் authenticate செய்வதற்கான password prompt-ஐ அமைக்கவும்.
3. db_name என்பது default database அமைக்கப்படுவதைக் காட்டுகிறது
4. பின் commands அடங்கியுள்ள file_name.sql எனும் file-ஐ server-க்கு அனுப்பவும்.
5. அடுத்தபடியாக, இந்த commands execute செய்யப்பட்டு அதன் results தெரிவிக்கப்படும்.

6. கடைசியாக exit செய்யப்படும்.

ஒருவேளை, MySQL command-line ஏற்கனவே பயன்பாட்டில் உள்ளதெனில், query-யைப் பின்வருமாறு அமைத்தால் போதுமானது.

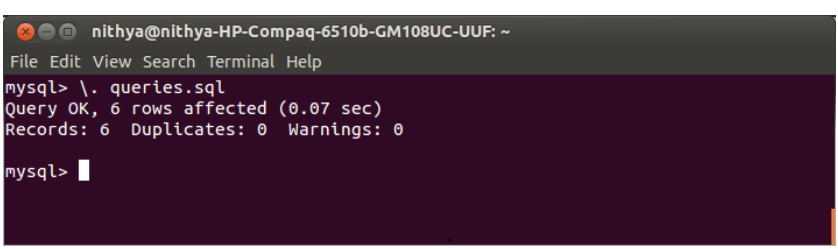
mysql> \. file_name.sql

இப்போது person எனும் table-ல், ‘batch mode’ மூலம் எவ்வாறு dataசெலுத்தப்படுகிறது என்று பார்க்கலாம். முதலில் queries-ஐ எல்லாம் ஒரு file-ல் enter செய்துவிட்டு, அதனை ‘queries.sql’ எனும் பெயரில் save செய்துகொள்ளவும். இது பின்வருமாறு.

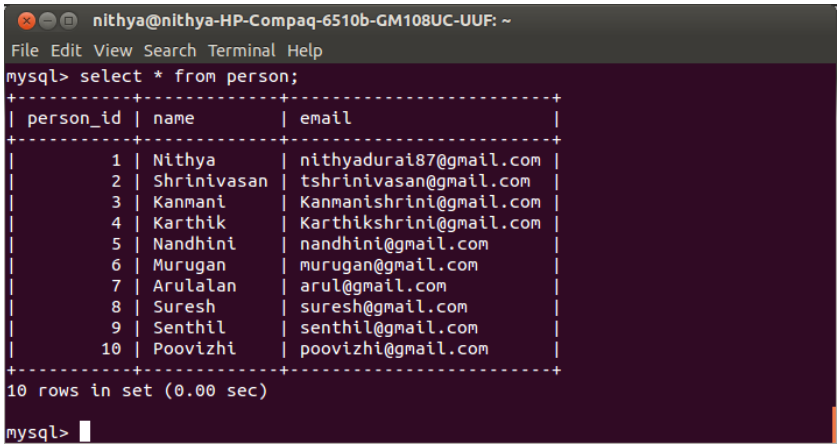


பின்னர் MySQL command line-ல், பின்வருமாறு enter செய்யவும்.

mysql> \. queries.sql



இது file-ல் இருக்கும் data-வை, table-ல் insert செய்துவிடும்.


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> select * from person;
person_id	name	email
1	Nithya	nithyadurai87@gmail.com
2	Shrinivasan	tshrinivasan@gmail.com
3	Kanmani	Kanmanishrini@gmail.com
4	Karthik	Karthikshrini@gmail.com
5	Nandhini	nandhini@gmail.com
6	Murugan	murugan@gmail.com
7	Arulalan	arul@gmail.com
8	Suresh	suresh@gmail.com
9	Senthil	senthil@gmail.com
10	Poovizhi	poovizhi@gmail.com
10 rows in set (0.00 sec)
mysql>

8.5 ஒரு table-ல் உள்ள data-வை மற்றொரு table-ல் insert செய்தல்

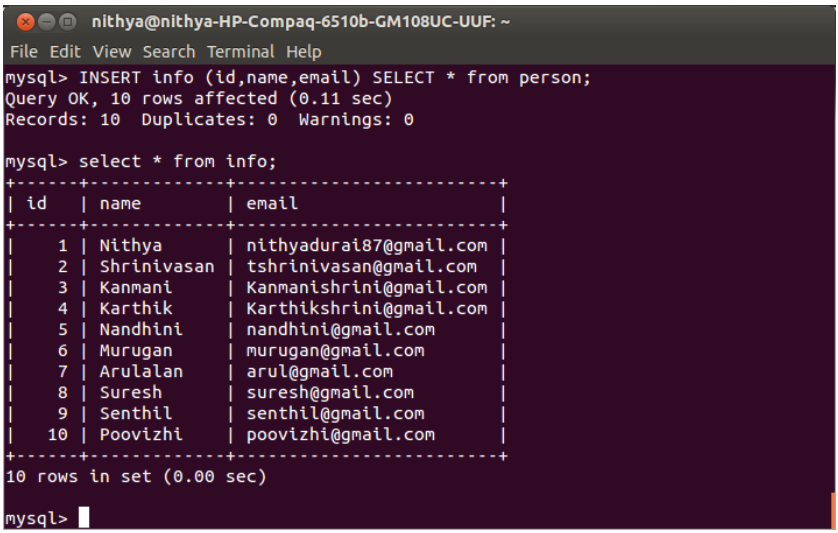
Insert command-ஆனது select command-வுடன் ஒன்றாக இணைந்து ஒரு

table-லிலிருந்து copy செய்யப்படும் rows-ஐ மற்றொரு table-க்குள் insert செய்யப் பயன்படுகிறது.
இந்த INSERT—-SELECT command-க்கான syntax மிகவும் சுலபமானது. Insert command-ன் முதல் பகுதியை (VALUES clause வரை) Select command-வுடன் இணைக்க வேண்டும். இது பின்வருமாறு.

INSERT table_one (list, of, columns) SELECT …;

உதாரணத்துக்கு, ‘person’ எனும் table-ல் இருக்கும் அனைத்து data-வையும், ‘info’ எனும் table-க்கு மாற்ற query-யைப் பின்வருமாறு அமைக்கலாம்.

INSERT info (id,name,email) SELECT * from person;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> INSERT info (id,name,email) SELECT * from person;
Query OK, 10 rows affected (0.11 sec)
Records: 10 Duplicates: 0 Warnings: 0
mysql> select * from info;
id	name	email
1	Nithya	nithyadurai87@gmail.com
2	Shrinivasan	tshrinivasan@gmail.com
3	Kanmani	Kanmanishrini@gmail.com
4	Karthik	Karthikshrini@gmail.com
5	Nandhini	nandhini@gmail.com
6	Murugan	murugan@gmail.com
7	Arulalan	arul@gmail.com
8	Suresh	suresh@gmail.com
9	Senthil	senthil@gmail.com
10	Poovizhi	poovizhi@gmail.com
10 rows in set (0.00 sec)
mysql>

இதில் Insert command-ல் இருக்கும் ஒவ்வொரு column-க்கும், select statement-ல் இருந்து ஒரு column, return செய்யப்படுகிறதா என்பதை உறுதி செய்து கொள்ளவும்.

8.6 ஒரு file-ல் உள்ள data-வை நேரடியாக table-ல் செலுத்துதல்

Data-வை பிற applications-லிருந்து MySQL-க்கு import செய்யும்போது, முதலில் அந்த data-வை ஒழுங்குபடுத்தப்பட்ட text format ஆக மாற்றி (Eg: tab-separated values) அதன்பின் MySQL-ல் import செய்ய வேண்டும்.

இவ்வாறு அல்லாமல் அந்த data-வை ஒரு set of SQL queries-ஆக மாற்றி அதன் பின்னரும் import செய்யலாம்.

ஆனால் இவ்வாறு செய்வது, அதிக நேரம் பிடிக்கக்கூடியதாகவும், மிகுதியான தவறுகளுக்கு நம்மை உட்படுத்தக்கூடியதாகவும் இருக்கும். எனவே LOAD DATA INFILE எனும் command, text files-ல் இருக்கும் வடிவமைக்கப்பட்ட data-வை MySQL-க்குள் செலுத்தப் பயன்படும் ஓர் எளிமையான வழியாகும். இதற்கான syntax பின்வருமாறு.

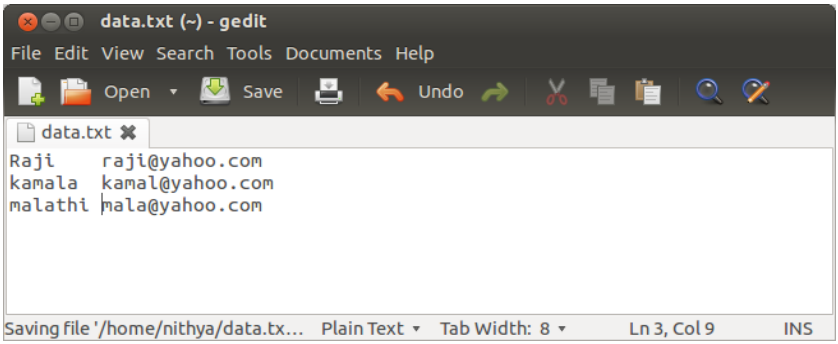
LOAD DATA INFILE ‘some_file.txt’

INTO TABLE ‘some_table’ (list, of, columns, …);

இந்த text file-க்குள் இருக்கும் data-வின் ஒவ்வொரு field-ம் ஒரு tab-ஆலும், ஒவ்வொரு row-ம் ஒரு புதிய line-ஆலும் ஒழுங்குபடுத்தப்பட்டிருக்கும்.

இந்த LOAD DATA INFILE எனும் command, இத்தகைய ஒழுங்குபடுத்தப்பட்ட file-ஐ எவ்வாறு import செய்வது என்பதை தானாகவே புரிந்துகொள்ளும் வகையில் இருக்கும். file-ல் இருக்கும் ஒவ்வொரு row-ம் இந்த command-ஆல் read செய்யப்பட்டு பின்னர் அதற்குரிய table-ல் column mapping-ஐ பயன்படுத்தி அந்த row செலுத்தப்படும்.

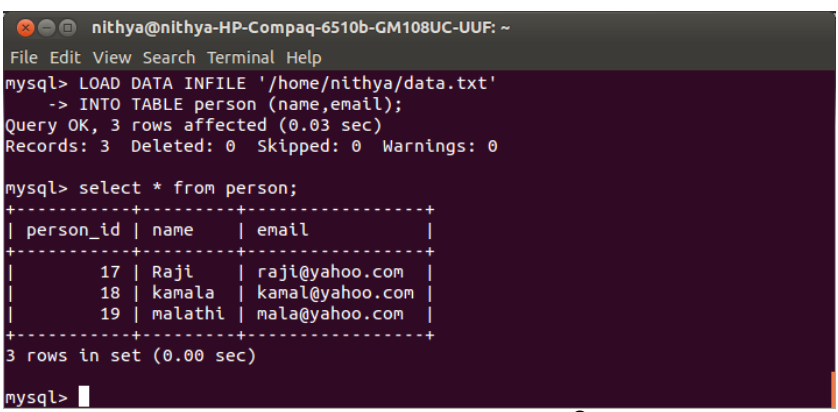
உதாரணத்துக்கு /home/nithya/data.txt எனும் இடத்தில் சேமித்துவைக்கப்பட்டுள்ள பின்வரும் சாரத்தை உள்ளடக்கிய இந்த file எவ்வாறு person எனும் table-க்கு மாற்றப்படுகிறது என்று பார்க்கலாம்.



இதனை upload செய்வதற்கான command பின்வருமாறு அமையும்.

LOAD DATA INFILE '/home/nithya/data.txt'

INTO TABLE person (name,email);
இந்த command, run செய்யப்பட்ட பின்னர், book எனும் table-ல், file-ல் உள்ள data அனைத்தும் செலுத்தப்பட்டிருப்பதை பின்வருமாறு காணலாம்.


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> LOAD DATA INFILE '/home/nithya/data.txt'
-> INTO TABLE person (name,email);
Query OK, 3 rows affected (0.03 sec)
Records: 3 Deleted: 0 Skipped: 0 Warnings: 0
mysql> select * from person;
person_id	name	email
17	Raji	raji@yahoo.com
18	kamala	kamal@yahoo.com
19	malathi	mala@yahoo.com
3 rows in set (0.00 sec)
mysql>

LOAD DATA INFILE எனும் command-ல் பின்வருவனவற்றை கவனிக்கவும்.

- MySQL server இயங்கும் அதே machine-ல் datafile இருக்க வேண்டும்.
- Data load செய்யப்படவேண்டிய file-ஐ குறிப்பிடும்போது அதன் பெயரை மட்டும்

குறிப்பிடாமல் அதன் முழு path-ஐயும் குறிப்பிட வேண்டும். இல்லையெனில், இந்த MySQL server குறிப்பிட்ட path-ல் தேடாமல், வேறு எங்காவது அதன் data directory-ல் தேடிக்கொண்டிருக்கும்
- Windows systems-ல் கூட நீங்கள் blackslashes-க்கு பதிலாக forward slash-ஐ பயன்படுத்தி path-ஐ குறிப்பிடலாம். உதாரணத்துக்கு unix-style path ('/home/nithya/data.txt') என்று குறிப்பிடப்படுகிறது என்றால், அது windows-ல் 'C:/Desktop/data.txt' என்று குறிப்பிடப்படும். இங்கு நீங்கள் blackslashes-ஐ பயன்படுத்தினால் அது character escape sequences-ஆக interpret செய்யப்படும்.

மேலும் பின்வரும் குறிப்புகள், இந்த load data infile -ன் பயன்பாட்டை இன்னும் சுலபமாக அமைக்கும்.

இந்த command-ன் output ஒரு warning அல்லது error-வுடன் வருகிறது எனில் (eg:

Query OK, 3 rows affected, 1 warning (0.01 sec))

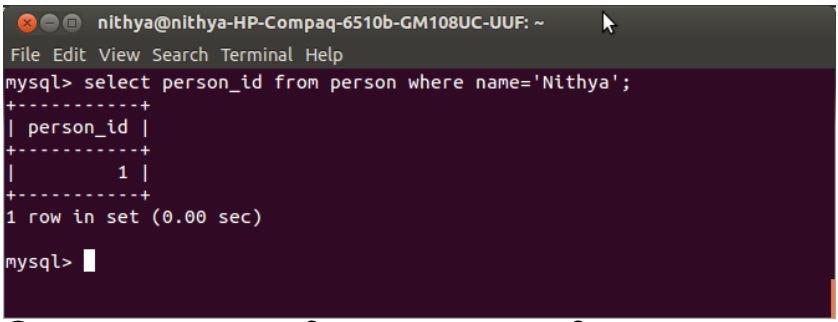
SHOW WARNINGS அல்லது SHOW ERRORS எனும் command-ஐப் பயன்படுத்தி அது எவ்வகையான பிரச்சனைகளைக் கொண்டுள்ளது என்று பார்க்கலாம்.

8.7 loan எனும் table-ல் data-வை insert செய்தல்

இதில் data-வை insert செய்வதற்கு நமக்கு book மற்றும் person எனும் இரண்டு table-ன் உதவியும் தேவை. அதாவது "நித்யா" எனும் நபர் "சில நேரங்களில் சில மனிதர்கள்" எனும் புத்தகத்தைப் பெற விரும்புகிறார் எனில் முதலில் இந்த நபர் மற்றும் புத்தகத்தின் primary key-யை நாம் தெரிந்து கொள்ள வேண்டும். அதன் பின்னர் தான் இந்த primary key-யைப் பயன்படுத்தி மதிப்புகளை loan எனும் table-ல் செலுத்த வேண்டும்.

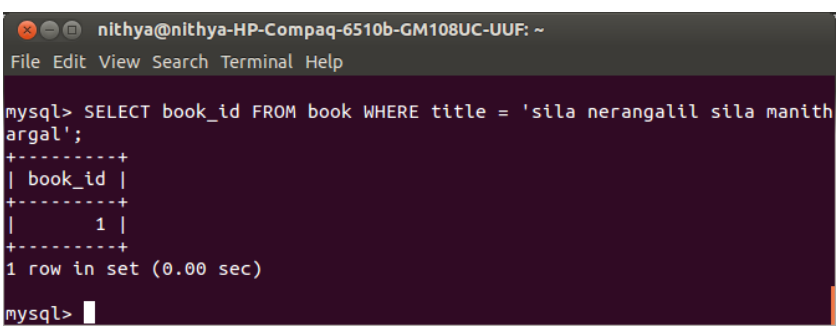
நித்யா எனும் நபரின் primary key-யைத் தெரிந்து கொள்ள, query-யைப் பின்வருமாறு அமைக்கவும்.

SELECT person_id FROM person WHERE name = ‘ Nithya’;



இந்த output மூலம் நித்யா எனும் நபரின் primary key ’1′ எனத் தெரியவந்துள்ளது.
அடுத்ததாக “சில நேரங்களில் சில மனிதர்கள்” எனும் புத்தகத்தின் primary key-யைத் தெரிந்து கொள்ள, query-யைப் பின்வருமாறு அமைக்கவும்.

SELECT book_id FROM book WHERE title = ‘sila nerangalil sila manithargal’;



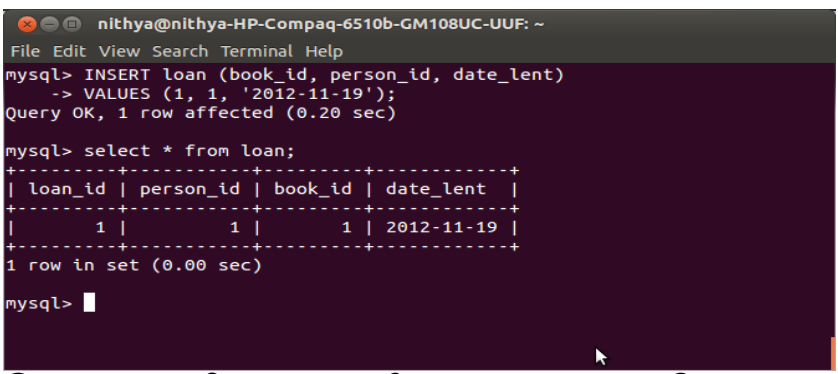
இந்த output மூலம் “சில நேரங்களில் சில மனிதர்கள்” எனும் புத்தகத்தின் primary key ’1′ எனத் தெரியவந்துள்ளது.

இப்போது இந்த இரண்டு மதிப்புகளையும்

வைத்துக்கொண்டு நித்யா எந்தத் தேதியில் “சில நேரங்களில் சில மனிதர்கள்” எனும் புத்தகத்தைப் பெற்றுள்ளார் எனும் விவரத்தை loan எனும் table-ல் பின்வருமாறு செலுத்தலாம்.

INSERT loan (book_id, person_id, date_lent)

VALUES (1, 1, ’2012-11-19′);



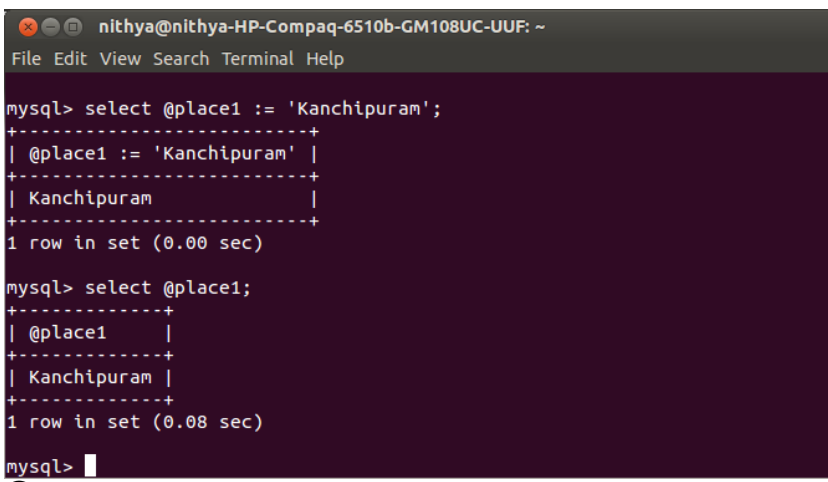
இவ்வாறு மதிப்புகளை நேரடியாக table-ல் செலுத்தாமல் அதனை ஒரு user variable-க்கு assign செய்துவிட்டு, அந்த user variable-ஐ நாம் query-ல் பயன்படுத்தலாம்.

ஒரு user variable-க்கான format, @name என்று இருக்கும். இந்த name எனும் இடத்தில் நாம் A-Z வரையிலான எழுத்துக்களையோ, 0-9 முடிய இருக்கும் எண்களையோ அல்லது $,-போன்ற குறிகளையோ பயன்படுத்தி ஒரு user variable-ஐ உருவாக்கலாம்.

இவ்வாறு உருவாக்கப்பட்ட ஒரு variable-க்கு := எனும் operator-ஐப் பயன்படுத்தி மதிப்புகளை assign செய்யலாம். உதாரணம்:

SELECT @place1 := ‘Kanchipuram’;
இதில் ‘Kanchipuram’ எனும் மதிப்பை உள்ளடக்கிய ஒரு ‘@place1’ எனும் user variable உருவாக்கப்பட்டுள்ளது. இப்போது பின்வரும் query-ஐ அமைத்து, run செய்தால் அந்த variable அதன் மதிப்பினை வெளிப்படுத்தும்.

SELECT @place1;



இவ்வாறாக நாம் loan எனும் table-ல் user variable-ஐப் பயன்படுத்தி , மதிப்புகளை பின்வருமாறு செலுத்தலாம். இவ்வாறு user variable மூலம் data-ஆனது table-க்கு செலுத்தப்படும்போது, தெரியாமல் ஏற்படும் தவறுகள் தவிர்க்கப்படுகின்றன.

SELECT @person_id := person_id FROM person

WHERE name = ‘Shrinivasan’;

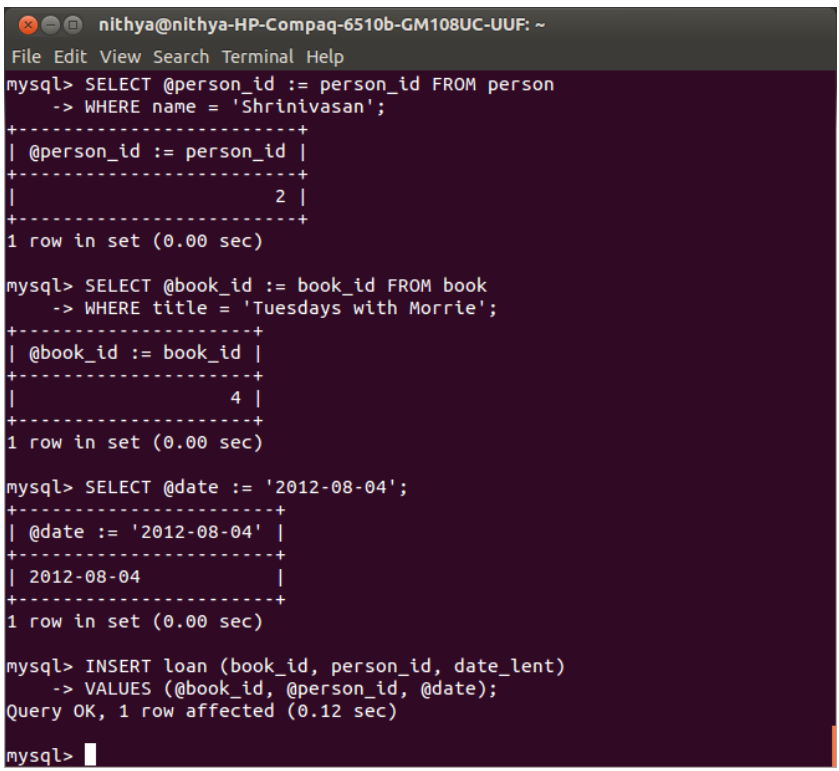
SELECT @book_id := book_id FROM book

WHERE title = ‘Tuesdays with Morrie’;

SELECT @date := ’2012-08-04’;

INSERT loan (book_id, person_id, date_lent)

VALUES (@book_id, @person_id, @date);



இங்கு தனித்தனியான select statement-ஐப் பயன்படுத்தி user variable உருவாக்கப்பட்டுள்ளது. இதனை ஒரே select statement-ஐப் பயன்படுத்தியும் நாம் பின்வருமாறு உருவாக்கி அதனை table-ல் insert செய்யலாம்.

SELECT @book_id := book_id,

@person_id := person_id,

@date := ’2012-12-12’

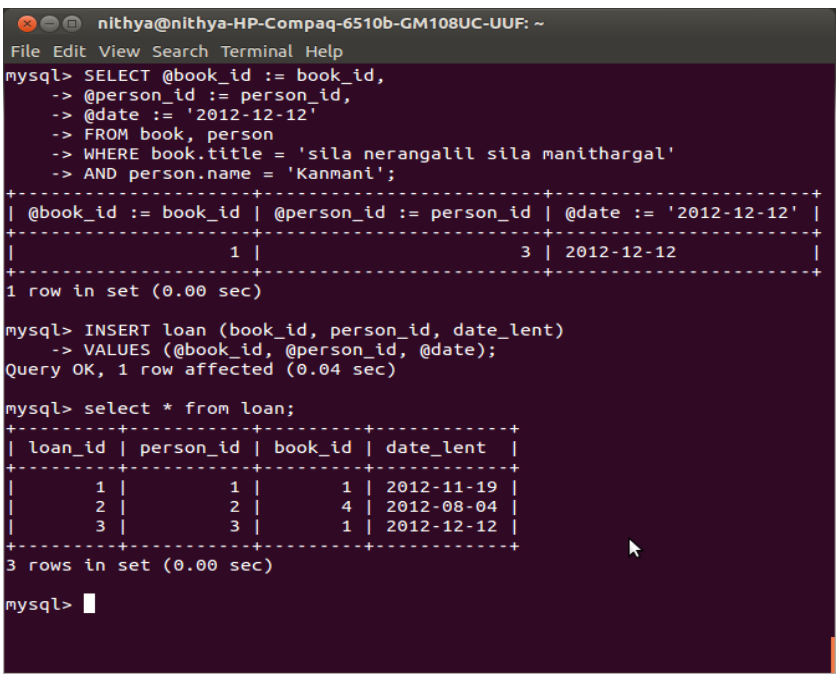
FROM book, person

WHERE book.title = ‘sila nerangalil sila manithargal’

AND person.name = ‘Kanmani’;

INSERT loan (book_id, person_id, date_lent)

VALUES (@book_id, @person_id, @date);


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SELECT @book_id := book_id,
-> @person_id := person_id,
-> @date := '2012-12-12'
-> FROM book, person
-> WHERE book.title = 'sila nerangalil sila manithargal'
-> AND person.name = 'Kanmani';
| @book_id := book_id | @person_id := person_id | @date := '2012-12-12' |
| 1 | 3 | 2012-12-12 |
1 row in set (0.00 sec)
mysql> INSERT loan (book_id, person_id, date_lent)
-> VALUES (@book_id, @person_id, @date);
Query OK, 1 row affected (0.04 sec)
mysql> select * from loan;
loan_id	person_id	book_id	date_lent
1	1	1	2012-11-19
2	2	4	2012-08-04
3	3	1	2012-12-12
3 rows in set (0.00 sec)
mysql>

இந்த இரண்டு queries-ஐயும்கூட நாம் user variables ஏதும் இல்லாமல் ஒரே query-யாக மாற்றி இன்னும் optimize செய்யலாம். இது பின்வருமாறு.

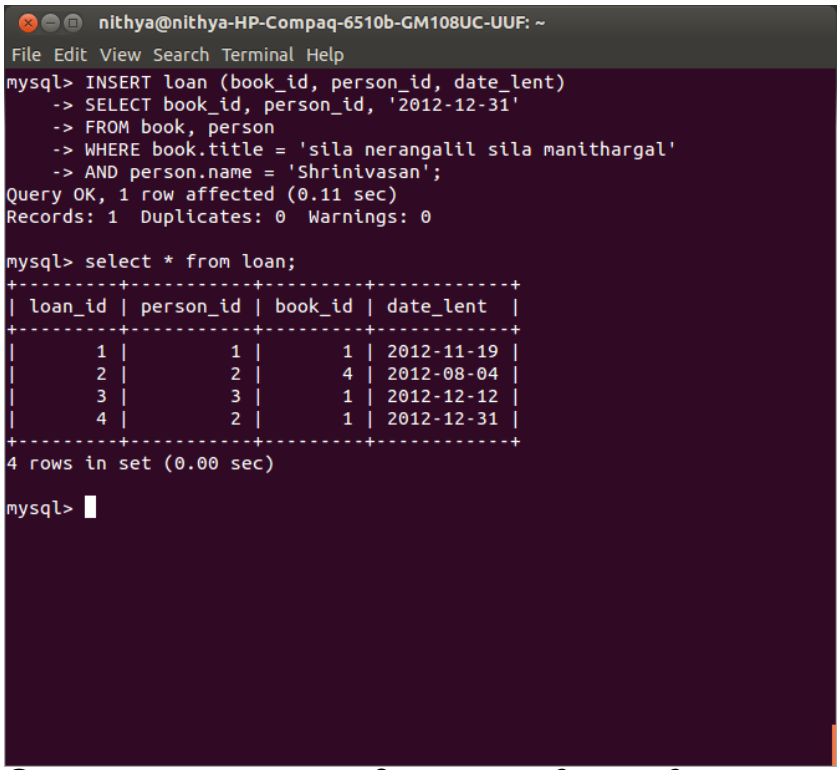
INSERT loan (book_id, person_id, date_lent)

SELECT book_id, person_id, ’2012-12-31’

FROM book, person

WHERE book.title = ‘sila nerangalil sila manithargal’

AND person.name = ‘Shrinivasan’;



இவ்வாறாக நாம் ஒரு விஷயத்தையே பல்வேறு வழிவகைகளில் வெவ்வேறாகச் செய்யலாம் என இப்போது கற்றுக் கொண்டோம். மேலும் ஒரு பெரிய query-யானது பல்வேறு எளிய queries-ஐக் கொண்டே உருவாக்கப்படுகிறது எனவும் தெரிந்து கொண்டோம்.

8.8 Function மூலம் date-ஐ insert செய்தல்

Now() எனும் function, ஒரு query-ல் அழைக்கப்படும்போது, அதன் MySQL server எந்த machine-ல் ஓடிக்கொண்டிருக்கிறதோ, அந்த machine-ன் date மற்றும் time-ஐ insert செய்யும். இதற்கான syntax பின்வருமாறு.

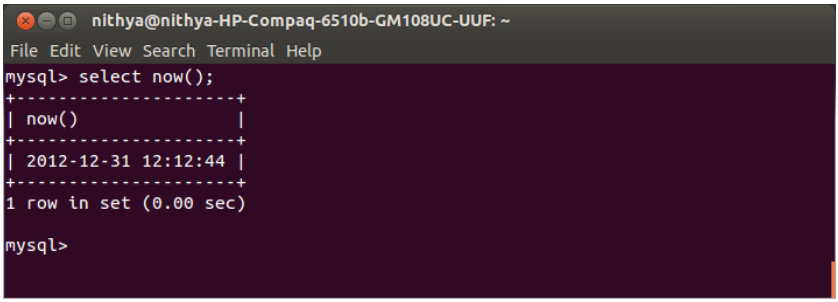
INSERT some_table (some_column) VALUES (NOW ());

மேலே இருக்கும் SQL-ல் பயன்படுத்தப்பட்டுள்ள Now ()- எனும் function தற்போதைய date மற்றும் time-ஐ கொடுக்கப்பட்டுள்ள column-ல் insert செய்யப்பயன்படும்.

பொதுவாக இந்த function return செய்யும் text-ஆனது தேவையான இடைவெளி மற்றும் punctuation-ஐப் பெற்று நாம் சுலபமாக படிக்கும் வகையில் அமையும். உதாரணத்துக்கு,

SELECT NOW();

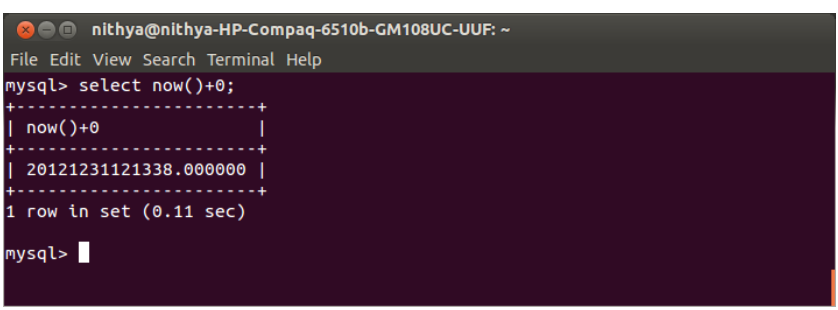
என வைத்து run செய்து பார்க்கவும். இது பின்வரும் format-ல் date மற்றும் time-ஐ return செய்யும்.



மேலும்,

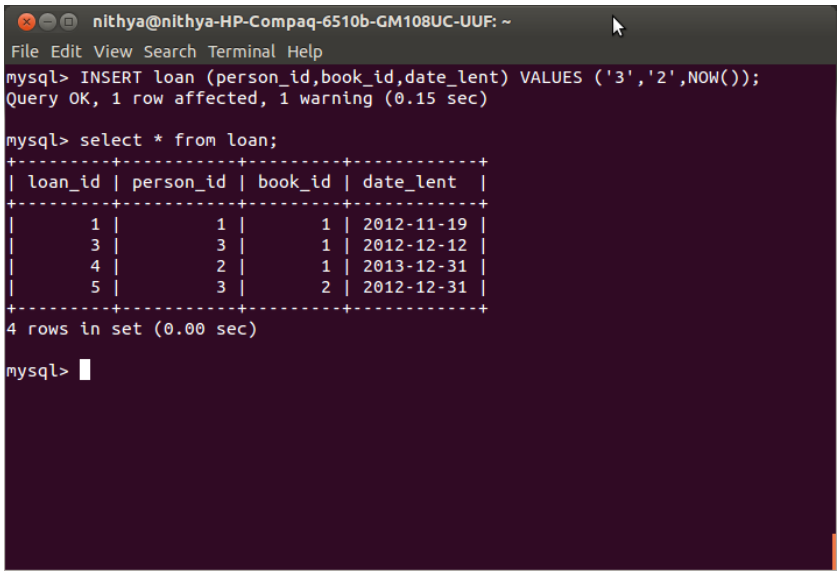
SELECT NOW() + 0;

என வைத்து run செய்யும்போது, இது பின்வரும் format-ல் date-ஐ return செய்யும்.



எனவே 'loan' எனும் table-ல், ஒரு சில புத்தகங்களுக்கு தற்போதைய தேதியை insert செய்ய விரும்பினால், now() எனும் function-ஐப் பின்வருமாறு பயன்படுத்தலாம்,

INSERT loan (person_id,book_id,date_lent) VALUES (’3′,’2′,NOW());


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> INSERT loan (person_id,book_id,date_lent) VALUES ('3','2',NOW());
Query OK, 1 row affected, 1 warning (0.15 sec)
mysql> select * from loan;
+---------+-----------+---------+------------+
| loan_id | person_id | book_id | date_lent |
+---------+-----------+---------+------------+
1	1	1	2012-11-19
3	3	1	2012-12-12
4	2	1	2013-12-31
5	3	2	2012-12-31
+---------+-----------+---------+------------+
4 rows in set (0.00 sec)
mysql>

9

Library எனும் மாதிரி Database-ல் இருக்கும் data-வை எடுத்தல்

இங்கு நாம் எவ்வாறு தகவல்களை table-லிலிருந்து எளிய queries-ஐப் பயன்படுத்திப் பெறுவது என்று பார்க்கலாம்.

பின்வரும் query ஒரு table-ல் இருக்கும் அனைத்து data-வையும் return செய்யும்.

SELECT * FROM book;

9.1 Limit Clause-ன் பயன்பாடு

LIMIT Clause-ஐப் பயன்படுத்தி நமக்கு வேண்டிய rows-ஐ மட்டும் கொடுக்குமாறு query-யை அமைக்கலாம். இது பின்வருமாறு.

SELECT author FROM book LIMIT 2;

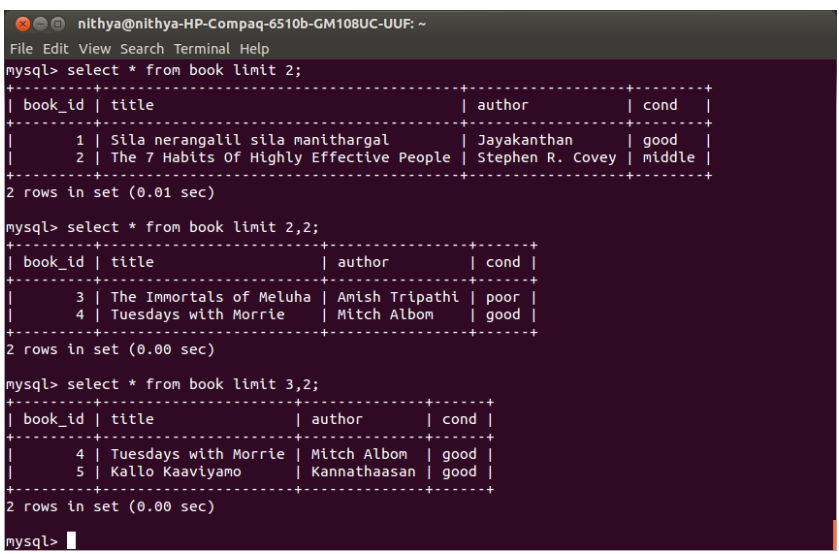
இது முதல் 2 rows-ஐ return செய்யும்.

SELECT author FROM book LIMIT 2, 2;

இது முதல் 2 rows-ஐ விட்டுவிட்டு, அடுத்த 2 rows-ஐ return செய்யும்.

SELECT author FROM book LIMIT 3, 2;

இது முதல் 3 rows-ஐ விட்டுவிட்டு, அடுத்த 2 rows-ஐ return செய்யும்.


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> select * from book limit 2;
book_id	title	author	cond
1	Sila nerangalil sila manithargal	Jayakanthan	good
2	The 7 Habits Of Highly Effective People	Stephen R. Covey	middle
2 rows in set (0.01 sec)
mysql> select * from book limit 2,2;
book_id	title	author	cond
3	The Immortals of Meluha	Amish Tripathi	poor
4	Tuesdays with Morrie	Mitch Albom	good
2 rows in set (0.00 sec)
mysql> select * from book limit 3,2;
book_id	title	author	cond
4	Tuesdays with Morrie	Mitch Albom	good
5	Kallo Kaaviyamo	Kannathaasan	good
2 rows in set (0.00 sec)
mysql>

இவ்வாறாக LIMIT Clause மூலம் நமக்கு வேண்டிய rows-ஐ மட்டும் நாம் பெற்றுக்கொள்ளலாம்.

9.2 Order by மூலம் result-ஐ வரிசைப்படுத்துதல்

MySQL-லிலிருந்து எடுக்கப்படும் rows வரிசைப்படுதப்பட்டோ அல்லது வரிசைப்படுத்தப்படமலோ இருக்கும். ஆனால் இந்தத் தகவல்கள் எவ்வாறு வரிசைப்படுத்தப்பட வேண்டும் என்பதை order by clause-ஐப் பயன்படுத்திக் கூறலாம். இந்த order by clause-ல் கொடுத்துள்ள column-ன் மதிப்புகளுக்கு ஏற்றவாறு table-ல் உள்ள தகவல்களெல்லாம் வரிசைப்படுத்தப்பட்டு வெளிவரும். இதைப் பின்வருமாறு காணலாம்.

SELECT * FROM book ORDER BY title;

'book' எனும் table-ல் இருந்து எடுக்கப்படும் அனைத்து records-ம், title-ன் அடிப்படையில் வரிசைப்படுத்தப்பட்டிருக்கும்.

மேலே கொடுக்கப்பட்டுள்ள code-க்கான விளக்கம் பின்வருமாறு.

- முதலில் பின்வரும் query, server-க்கு செலுத்தப்படும்.

SELECT * FROM book ORDER BY title;

- இந்த query, return செய்யும் ஒவ்வொரு row-வின் title-ம் ஒரு list-க்குள் செலுத்தப்படும். இது பின்வருமாறு.

Sila nerangalil sila manithargal (Row 1)

The 7 Habits of Highly Effective People (Row 2)

The Immortals of Meluha (Row 3)

Tuesdays with Morrie (Row 4)

Kallo kaaviyamo (Row 5)

Thanneer thesam (Row 6)

Yen Yetherku Yeppadi (Row 7)

மேலும் இந்த title எந்த row-வுடன் தொடர்புடையது என்பது server-க்கு தெரிந்திருக்கும்.

- பின்னர் இந்த list-ஆனது alphabetical முறைப்படி வரிசைப்படுத்தப்பட்டு பின்வருமாறு அமையும். இது case-sensitive கிடையாது.

Kallo kaaviyamo (Row 5)

Sila nerangalil sila manithargal (Row 1)

Thanneer thesam (Row 6)

The 7 Habits of Highly Effective People (Row 2)

The Immortals of Meluha (Row 3)

Tuesdays with Morrie (Row 4)

Yen Yetherku Yeppadi (Row 7)

- இந்த வரிசைப்படுத்தப்பட்ட list-ன் முறைப்படி, அனைத்து rows-ம் வெளிவரத் தொடங்கும்.

9.3 DISTINCT மூலம் ஒரே மாதிரியான data-வை ஒரு முறை மட்டும் வெளிக்காட்டுதல்

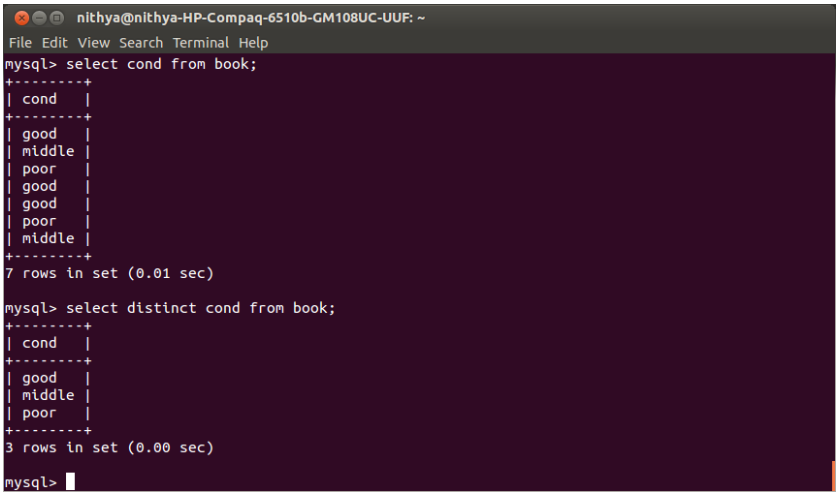
Distinct எனும் keywordஒரு data-வை ஒரே ஒரு முறை மட்டும் வெளிக்காட்டப் பயன்படுகிறது.

உதாரணத்துக்கு ‘book’ எனும் table-ல் இருக்கும் ‘cond’ எனும் column-ஐ, ‘distinct’ keyword இல்லாமல் வெளிப்படுத்தும் போது அது அனைத்து data-வையும் வெளிப்படுத்தும். அதையே நாம் ‘distinct’ எனக் கொடுத்து வெளிப்படுத்தும்போது ஒரு data-வை ஒருமுறை மட்டும் தான் வெளிப்படுத்தும்.

இது பின்வருமாறு.

SELECT cond FROM book;

SELECT DISTINCT cond FROM book;



இங்கு உள்ள முதல் query, ‘cond’ எனும் column-ல் இருக்கும் data-ஐ பல rows -ல் இருந்து எடுத்து, ஒவ்வொரு முறையும் அதே data-வை return செய்யும்.

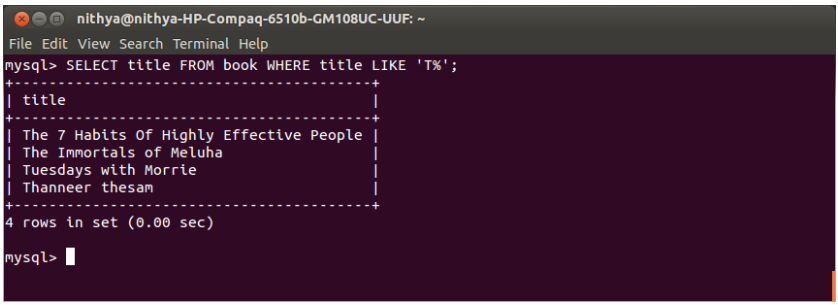
அடுத்ததாக உள்ள query-ல் select-ஐ தொடர்ந்து distinct எனும் clause -ஐ சேர்த்ததால், அது ஒரே மாதிரியான data-ஐ ஒரே ஒருமுறை மட்டும் return செய்துவிட்டு மற்ற duplicate data-ஐ புறக்கணித்து விட்டது.

9.4 ஒரே மாதிரியான **data**-வை **‘like’** மூலம் கண்டுபிடித்து வெளிக்காட்டுதல்

Like operator-ஐப் பயன்படுத்தி ஒரே மாதிரியான pattern-ல் இருக்கும் பல சொற்களைக் கண்டுபிடிக்கலாம். உதாரணத்துக்கு book-எனும் table-ல் இருந்து, T-ல்

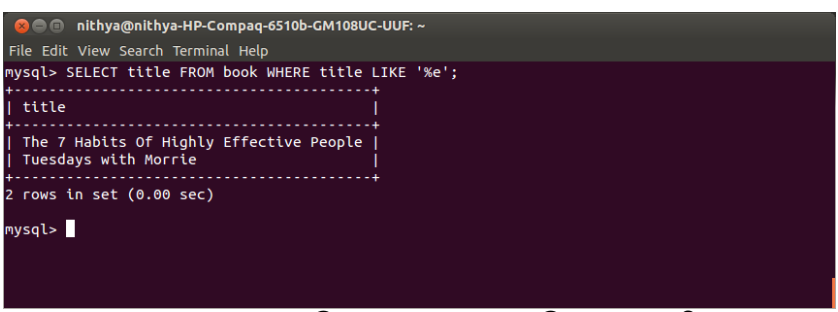
தொடங்கும் அனைத்து புத்தகத்தின் பெயர்களையும் பட்டியலிட, query-யைப் பின்வருமாறு அமைக்கலாம்.

SELECT title FROM book WHERE title LIKE ‘T%’;



அதேபோல் eஎனும் எழுத்தில் முடியும் அனைத்து புத்தகத்தின் பெயர்களையும் பட்டியலிட, query-யைப் பின்வருமாறு அமைக்கலாம்.

SELECT title FROM book WHERE title LIKE ‘%e’;



Like operator ஆனது சொற்களைப் பொருத்திப் பார்ப்பதற்காகப் பயன்படுத்தும் இரண்டு wildcard characters பின்வருமாறு.

- %(Percent) – இது 0அல்லது அதற்கு மேற்பட்ட எந்த வகையான characters-வுடனும் பொருந்தும்.

- _(Underscore) – இது எந்த வகையான character-ஆக இருந்தாலும் ஒரே ஒரு character-ல் தான் பொருந்தும்.

உதாரணத்துக்கு நீங்கள் book-எனும் table-ல் இருந்து S எனும் எழுத்தின் தொடர்ச்சியாக, குறைந்தது 8 எழுத்தாவது இருப்பது போன்ற author-ன் பெயர்களைப் பட்டியலிட query-யைப் பின்வருமாறு அமைக்க வேண்டும்.

SELECT author FROM book

WHERE author LIKE ‘S_____%’;

இங்கு ‘Sujatha’ எனும் author-ன் பெயர் வெளிப்படாததை கவனிக்கவும். ஏனெனில் இது 7 எழுத்து கொண்ட பெயர் ஆகும்.

9.5 Aggregate functions மூலம் data-வைக் கொண்டு கணக்கீடுகள் செய்தல்

MySQL-ல் ஒரு column-ல் இருக்கும் மதிப்புகளைக் கொண்டு கணக்குகள் போடுவதற்கோ அல்லது அதனை group செய்வதற்கோ பல functions உள்ளன.

min() எனும் function ஒரு column-ல் சேமித்து வைக்கப்பட்டிருக்கும் மதிப்புகளில், மிகச்சிறிய மதிப்பினைக் கண்டுபிடிக்கப்பயன்படும்.
max() எனும் function ஒரு column-ல் சேமித்து வைக்கப்பட்டிருக்கும் மதிப்புகளில் மிகப்பெரிய மதிப்பினைக் கண்டுபிடிக்கப் பயன்படும்.
அதுபோலவே avg() எனும் function அந்த column-ல் சேமித்து வைக்கப்பட்டிருக்கும் அனைத்து மதிப்புகளின் average மதிப்பினைக் கண்டுபிடித்து அதனை தசம எண்ணில் வெளிப்படுத்தும்.

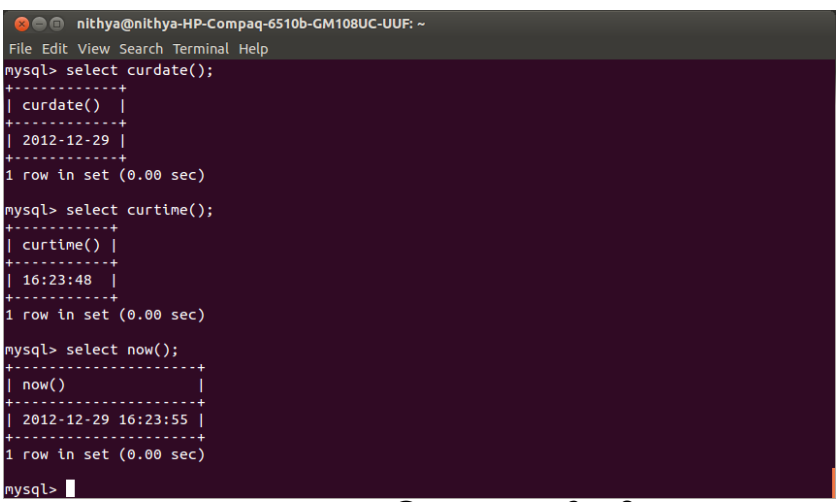
இதனை பின்வரும் உதாரணத்தில் காணலாம்.

SELECT MIN(marks), AVG(marks), MAX(marks) FROM person;

9.6 Date functions மூலம் தேதி மற்றும் நேரத்தினைக் கையாளுதல்

Date மற்றும் time-ஐ ஒப்பீடு செய்து அதனைக் கையாளுவதற்கு MySQL-ல் பல functions உள்ளன. MySQL-ஆனது தேதி மற்றும் நேரம் பற்றிய தகவல்களை பின்வரும் இரண்டு வடிவத்தில் குறிப்பிடுகின்றது.

- YYYY-MM-DD HH:MM:SS (Eg: '2005-09-30 18:43:01')
- YYYYMMDDHHMMSS (Eg: 20050930184301)



CURDATE() எனும் function இன்றைய தேதியையும்,

CURTIME() எனும் function தற்போதைய நேரத்தையும், அதேபோல்

now() எனும் function தற்போதைய தேதி மற்றும் நேரத்தையும் கொடுக்கவல்லது. இது பின்வருமாறு.

DATE_ADD() மற்றும் DATE_SUB() என்பது போன்ற functions

தேதிகளை வைத்து கணக்குப் போடுவதற்கோ அல்லது date/time-லிருந்து நேரம் அல்லது வருடம் போன்ற தகவல்களை மட்டும் பிரித்தெடுப்பதற்கோ பயன்படுகின்றன.

மேலும் கணித ஒப்பீட்டு குறிகளான, greater than(>),less than(<) மற்றும் equal to(=) போன்றவை தேதிகளை ஒன்றுடன் ஒன்று ஒப்பீடு செய்யப் பயன்படுகின்றன.

SELECT * FROM loan

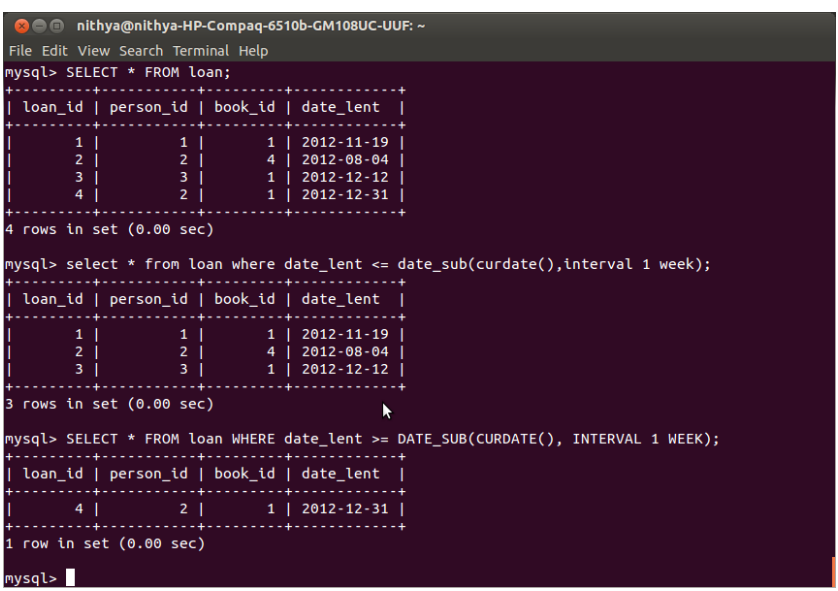
WHERE date_lent < =

DATE_SUB(CURDATE(), INTERVAL 1 WEEK);

SELECT * FROM loan

WHERE date_lent > =

DATE_SUB(CURDATE(), INTERVAL 1 WEEK);


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SELECT * FROM loan;
+---------+-----------+---------+------------+
| loan_id | person_id | book_id | date_lent |
+---------+-----------+---------+------------+
1	1	1	2012-11-19
2	2	4	2012-08-04
3	3	1	2012-12-12
4	2	1	2012-12-31
+---------+-----------+---------+------------+
4 rows in set (0.00 sec)
mysql> select * from loan where date_lent <= date_sub(curdate(),interval 1 week);
+---------+-----------+---------+------------+
| loan_id | person_id | book_id | date_lent |
+---------+-----------+---------+------------+
1	1	1	2012-11-19
2	2	4	2012-08-04
3	3	1	2012-12-12
+---------+-----------+---------+------------+
3 rows in set (0.00 sec)
mysql> SELECT * FROM loan WHERE date_lent >= DATE_SUB(CURDATE(), INTERVAL 1 WEEK);
+---------+-----------+---------+------------+
| loan_id | person_id | book_id | date_lent |
+---------+-----------+---------+------------+
| 4 | 2 | 1 | 2012-12-31 |
+---------+-----------+---------+------------+
1 row in set (0.00 sec)
mysql>

மேலே கொடுக்கப்பட்டுள்ள எடுத்துக்காட்டில் இருக்கும் முதல் query-ஆனது loan எனும் table-ல், தற்போதைய தேதியிலிருந்து அதன் ஒரு வார காலத்திற்கு முன்னால் வரை உள்ள data-ஐ பட்டியலிடப் பயன்படும்.

இரண்டாவதாக உள்ள query-ஆனது loan எனும் table-ல், தற்போதைய தேதியிலிருந்து அதன் ஒரு வார காலத்திற்கு முன்னால் வரை உள்ள data தவிர மற்ற data-ஐ பட்டியலிடப் பயன்படும்.

Date மற்றும் time functions-ஐ பயன்படுத்துவது சுலபமாகவே இருந்தாலும், உங்களுடைய program-ல் அதை நீங்கள் பயன்படுத்தும்போது வெவ்வேறு வகையான time zones, daylight savings time, wandering system clocks மற்றும் இரண்டு இலக்கத்தில் நூற்றாண்டுகளைக் குறிப்பிடுவது போன்றவற்றைப்பற்றி நினைவில் கொள்வது நல்லது.
பின்வரும் குறிப்புகள் இதுபோன்ற பிரச்சனைகளில் இருந்து உங்களைப் பாதுகாக்கப் பயன்படும்.

- Network Time Protocol(NTP) -ஐப் பயன்படுத்துவது,உங்கள் கணிப்பொறியின் நேரத்தை தற்போதைய நேரத்திற்கு அமைக்க உதவும்.
- தேதியை அமைக்கும்போது, முடிந்தவரை மிகத் துல்லியமான முறையில் அமைப்பது நல்லது.
- நேரத்தை நீங்கள் துல்லியமான முறையில் பார்க்க விரும்பினால், அதனை Greenwich Mean Time(GMT) எனப்படும் Coordinated Universal Time(UTC)-ல் சேமித்து வைக்கவும். பின்னர் உள்ளூர் நேரத்திற்கு ஏற்றவாறு காட்டும் logic-ஐ இதற்கான program-ல் இணைத்து விடவும்..

9.7 **Decimal columns** மூலம் தசம

எண்களை மிகத்துல்லியமாக சேமித்தல்

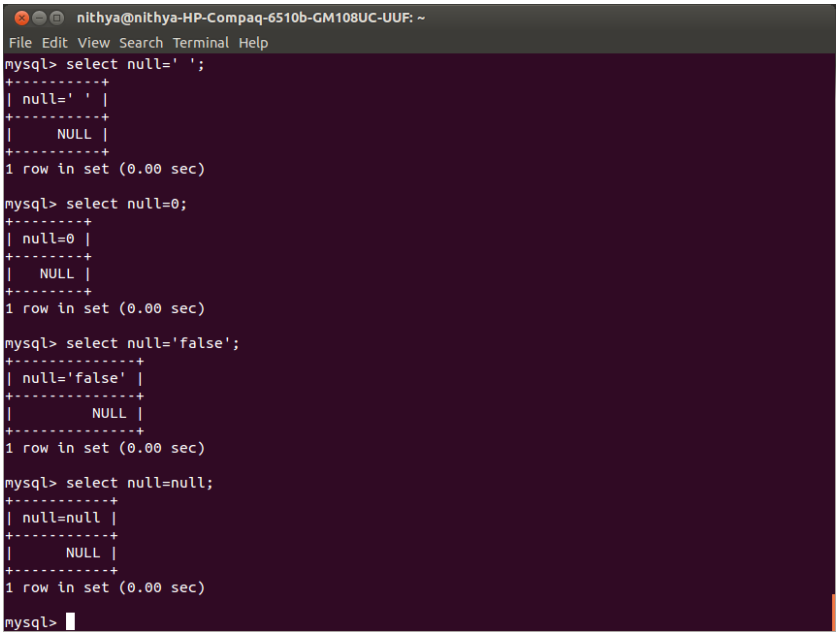
MySQL-ஆனது தசம எண்களை மிகத்துல்லியமான முறையில் சேமிக்க decimal வகை columns-ஐப் பயன்படுத்துகின்றன. இந்த decimal columns சேமிக்கப்படும் தசம எண்களை floating point numbers-ஆக அல்லாமல் strings-ஆக சேமிக்கின்றன. இந்த முறையில், தசம எண்களை சேமிப்பதற்கு நிறைய இடம் எடுத்துக் கொள்ளப்பட்டாலும், சேமிக்கப்படும் data எந்த அளவு துல்லியமாக இருக்க வேண்டும் என்பதைக் குறிப்பிட்டுக் கூற முடியும்.

மேலும் ஒரு decimal column-ல் எந்த அளவு data சேமிக்கமுடியும் என்பது அந்த MySQL-ன் version-ஐப் பொறுத்து மாறுபடும். அதற்கு MySQL-ன் manual-ஐப் பார்க்கவும். பொதுவாக அனைத்து versions-களோடும் ஒத்துப்போக தசம எண்ணின் வலப்புறம் இருக்கும் இலக்கங்களை 30 அல்லது அதற்கும் குறைவாக நிறுத்திக்கொள்வது நல்லது.

9.8 NULL-ன் பயன்பாடு

இந்த NULL என்பது எதையும் குறிக்காது. அதாவது empty அல்லது zero-வைக் கூட குறிக்காது. உதாரணத்துக்கு பின்வரும் queries-ஐ ஓட்டிப்பார்க்கவும்.

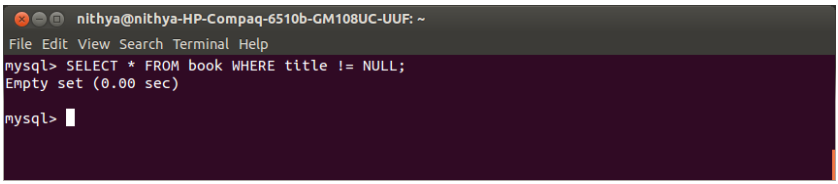
```
SELECT NULL = FALSE;

SELECT NULL = ”;

SELECT NULL = 0;
```



SELECT NULL = NULL;
அனைத்து queries-ன் விடையும் null என்பதே ஆகும். இந்த null என்பதற்கு எந்த ஒரு மதிப்பும் கிடையாது மற்றும் அது எந்த ஒரு data வகையையும் சார்ந்தது அல்ல. எனவே இந்த null-வுடன் boolean மதிப்பு false-ஐ ஒப்பிட்டாலும், empty string-ஐ ஒப்பிட்டாலும், 0 எனும் integer மதிப்புடன் ஒப்பிட்டாலும் அல்லது வேறு ஒரு null-வுடனே ஒப்பிட்டாலும் கூட, அது வெறும் null எனும் மதிப்பையே கொடுக்கும். வேறு எந்த ஒரு மதிப்பையும் கொடுக்காது.
நீங்கள் null மதிப்புகளைக் கொண்ட column-ஐ கையாளும்போது பின்வருவனவற்றை நினைவில் கொள்ளவும்.

- ஒரு column-ஐ null மதிப்புடன் ஒப்பீடு செய்யும்போது சாதாரண comparison operators-ஐ பயன்படுத்துவது எந்த ஒரு தகவலையும் பெற உதவாது. உதாரணத்துக்கு பின்வரும் query எந்த ஒரு rows-ம் return செய்யாது.

SELECT * FROM book WHERE title != NULL;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SELECT * FROM book WHERE title != NULL;
Empty set (0.00 sec)
mysql>

- எனவே எப்போதும் null மதிப்புடன் எதையாவது ஒப்பிடும்போது, Special Null Safe Operators எனப்படும் ISNULL மற்றும் < = > -ஐபயன்படுத்துவதே சிறந்தது.

9.9 Query result-ஐ ஒரு file-க்குள் அனுப்புதல்

இதனை INTO OUTFILE மற்றும் INTO DUMPFILE எனும் இரண்டு clause-ஐப் பயன்படுத்தி செய்யலாம்.

1. **INTO OUTFILE**

INTO OUTFILE எனும் clause-ஐ select statement-வுடன் பயன்படுத்தி, அந்த query மூலம் பெறும் மதிப்புகளை ஒரு file-க்குள் அனுப்ப முடியும்.

பொதுவாக இந்த clause-வுடன் simple select statement பயன்படுத்தப்படும்போது அதன் செயல்முறை எவ்வாறு அமையுமெனில், select query-லிருந்து பெறப்படும் மதிப்புகளை, INTO OUTFILE-ஐத் தொடர்ந்து

கொடுக்கப்பட்டுள்ள file-ல் tab separated format-ல் எழுதிவிடும். இந்த file-ல் எழுதும்போது, ஒவ்வொரு field-ம் ஒரு tab-ஆகவும், ஒவ்வொரு row-வும் ஒரு புதிய வரியாலும் வேறுபடுத்தி காட்டப்பட்டிருக்கும்.

SELECT * FROM book

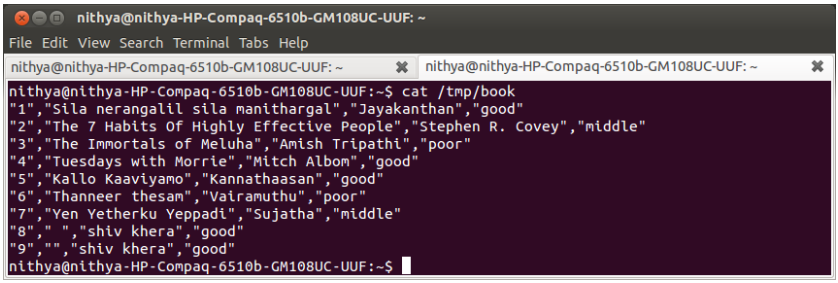
INTO OUTFILE '/tmp/book'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n';

மேலே உள்ள எடுத்துக்காட்டில் இருக்கும் query, CSV file-ஐ உருவாக்கும். ஏனெனில் எந்த format-ல் மதிப்புகள் எழுதப்பட வேண்டும் என்பது அந்த query-ல் தெளிவாகக் கொடுக்கப்பட்டுள்ளது. Character sequences எனப்படும் '\n' மற்றும் '\t' என்பது முறையே ஒரு புதிய line-ஆகவும் மற்றும் tab-ஆகவும் MySQL server-ஆல் interpret செய்து புரிந்து கொள்ளப்படும்.



1. **INTO DUMPFILE**

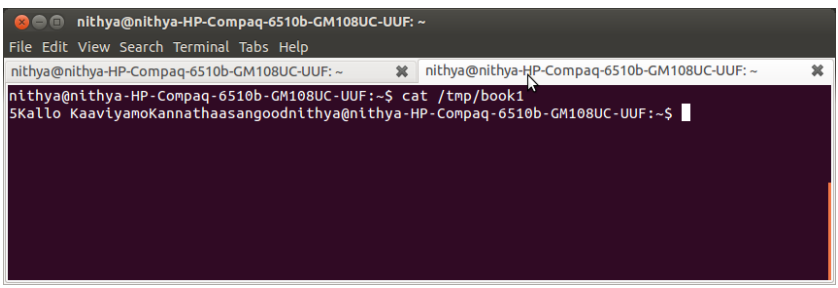
இந்த INTO DUMPFILE clause-ஐ பயன்படுத்துவதன் மூலம் select statement-லிருந்து பெறப்படும் மதிப்பு எந்த ஒரு ஒழுங்குதலும் செய்யப்படாமல் அப்படியே கொடுக்கப்பட்டுள்ள file-க்கு அனுப்பப்படுகிறது. இந்த INTO DUMPFILE எனும் clause ஒரே ஒரு row-வை return செய்யும் select statement-வுடன் மட்டுமே பயன்படுத்தப்படும். இந்த select statement-ல் ஒன்றுக்கு மேற்பட்ட columns கொடுக்கப்பட்டிருப்பின் அதன் மதிப்புகள் ஒன்று சேர்க்கப்பட்டு எந்த ஒரு வேறுபாடும் இல்லாமல் மொத்தமாக file-க்குள் எழுதப்படும். இது பின்வருமாறு.

SELECT * FROM book

WHERE book_id =5

INTO DUMPFILE “/tmp/book1″;

இது தொடர்ச்சியாக எந்த ஒரு இடைவெளியும் இல்லாமல் எழுதப்பட்டிருக்கும் ஒரு file-ஐ உருவாக்கும்.



10

Library database-ல் இருக்கும் பல்வேறு **Tables**-ஐ இணைத்தல்

எவ்வாறு எளிய queries-ஐப் பயன்படுத்தி நமக்கு வேண்டிய தரவுகளை ஒரு table-ல் இருந்து எடுப்பது என்பதைப் பற்றி ஏற்கனவே பார்த்துவிட்டோம். இந்தப் பாகத்தில், sql-ன் ஒருசில வலிமைமிகு அங்கங்களான Joins, Subqueries மற்றும் Union போன்றவற்றைப் பயன்படுத்தி மேம்பட்ட queries-ஐ உருவாக்கி, அதன் மூலம் தரவுகளை எவ்வாறு ஒன்றுக்கும் மேற்பட்ட tables-ல் இருந்து எடுப்பது என்று பார்க்கலாம்.

10.1 இணைப்புகள் உருவான விதம்

Relational databases-ன் ஒரு சிறப்பு அம்சம் என்னவெனில், அது data-வை ஒன்றுக்கும் மேற்பட்ட tables-ல் ஓர் ஒழுங்குபடுத்தப்பட்ட முறையில் சேமித்து வைத்திருக்கும். பொதுவாக ஒரு table-ல் ஒவ்வொரு row-வையும் தனித்தனியாக அடையாளம் கண்டுபிடிக்க உதவும் column-ஐ அந்த table-ன் primary key என்கிறோம். அதேபோல் ஒரு table-ல் இருக்கும் primary key அல்லாத மற்றொரு column, வேறொரு table-ன் primary key-யோடு ஒத்து இருக்கிறது எனில், அது அந்த table-ன் foreign key எனப்படும்.
உதாரணத்துக்கு book எனும் table-ல் book_id என்பது primary key எனில், இந்த column அந்த table-ல் இருக்கும் ஒவ்வொரு row-க்கும் ஒரு தனி எண் மதிப்பினைக் கொண்டிருக்கும். அதேபோல், person எனும் table-ல் இருக்கும் person_id எனும் column அந்த table-ல் இருக்கும் ஒவ்வொரு row-க்கும் ஒரு தனி எண் மதிப்பினைக் கொண்டிருக்கும்.

இப்போது மூன்றாவதாக இருக்கும் loan எனும் table-ல் “எந்தெந்த புத்தகங்கள் எந்தெந்த நபருக்கு வழங்கப்பட்டுள்ளன” எனும் விவரங்களைச்

சேகரிப்பதற்காக book மற்றும் person எனும் இரண்டு table-ஐயும் தொடர்பு கொள்ளும். இவ்வாறு தொடர்பு கொள்வதற்காக இந்த loan எனும் table-ல் இடம் பெற்றிருக்கும் book_id மற்றும் person_id எனும் columns, இந்த table-ன் foreign key எனப்படும்.

எனவே ஒரு புத்தகம் ஒரு நபருக்கு வழங்கப்படும்போது, அந்த புத்தகத்தின் முழு விவரங்களையும் நாம் loan எனும் table-ல் மீண்டும் செலுத்தாமல், வெறும் book_id எனும் column-ன் உதவியுடன் அந்த விவரங்களை வேறொரு book எனும் table-ல் இருந்து பெற்றுக்கொள்ளலாம்.

அவ்வாறே புத்தகம் பெற்றுக்கொண்ட நபரின் முழு விவரங்களையும், loan எனும் table-இல் இருக்கும் person_id உதவிகொண்டு person எனும் table-ல் இருந்து பெற்றுக்கொள்ளலாம். எனவே loan table-ல் நமக்கு வேண்டிய இன்னும் ஒன்றே ஒன்று தான். இந்த transaction எந்த தேதியில் நடைபெற்றுள்ளது என்பதைத் தெரிந்து கொள்ள date_lent எனும் column இருந்தால் போதுமானது. இவ்வாறாக loan,person மற்றும் book எனும் மூன்று tables-ம் ஒன்றுடன் ஒன்று இணைக்கப்பட்டுள்ளன.

10.2 Where clause மூலம் tables-ஐ இணைத்தல்

இரண்டு அல்லது அதற்கு மேற்பட்ட tables-லிருந்து data-வை எடுக்க வேண்டுமெனில், அந்த tables-ஐ primary key மற்றும் foreign key உதவியுடன் இணைக்க வேண்டும். பொதுவாக இது எவ்வாறு செய்யப்ப்படுமெனில், எந்தெந்த tables-லிருந்து data-வை எடுக்க வேண்டுமோ அந்த tables-ஐ எல்லாம் select statement-ன் from clause-ல் பட்டியலிடவேண்டும். பின்னர் where clause-ஐப்

பயன்படுத்தி அந்த tables-க்கு இடையிலிருக்கும் primary key-foreign key தொடர்பை விவரிக்கவேண்டும்.
இதற்கான syntax பின்வருமாறு.

SELECT columns

FROM table1, table2

WHERE table1.foreign_key = table2.primary_key;
இங்கு where clause-ஐப் பயன்படுத்துவது மிக முக்கியமானது. ஏனெனில், இது குறிப்பிடப்படவில்லை எனில், முதல் table-ல் இருக்கும் ஒவ்வொரு row-வும் , அடுத்த table-ல் இருக்கும் ஒவ்வொரு row-வுடன் இணைக்கப்பட்டு நமக்குப் பயன்படாத வகையில் result வெளிவரத் தொடங்கும்.
பின்வரும் எடுத்துக்காட்டில் உள்ள query-ல், loan, person மற்றும் book எனும் மூன்று tables-ம் இணைக்கப்பட்டு, அதிலிருந்து சில பொதுவான தரவுகள் தேர்ந்தெடுக்கப்பட்டுள்ளன.

SELECT loan.date_lent, person.name, book.title

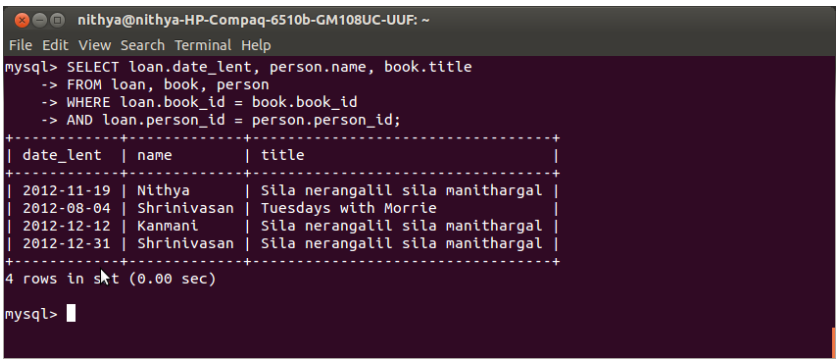
FROM loan, book, person

WHERE loan.book_id = book.book_id

AND loan.person_id = person.person_id;

இந்த query-ன் select statement-ல் இருக்கும் ஒவ்வொரு column-க்கும் முன்னால் உள்ள prefix, அந்த column எந்த table-லிலிருந்து எடுக்கப்பட வேண்டும் என்பதைக் குறிப்பிடும் qualifier ஆகும். ஏனெனில் ஒரே பெயர் கொண்ட columns இரண்டு tables-ல் பயன்படுத்தப்படும்போது, அந்த column-ஐ எந்த table-லிலிருந்து எடுக்க வேண்டும் என்பது புரியாமல் MySQL-ஆனது error-ஐ வெளிப்படுத்தும்.
எனவே இதனைத் தவிர்ப்பதற்காக ஒவ்வொரு column-க்கும் முன்னால் அதன் table name-ஐ qualifier-ஆக

குறிப்பிடுவது நல்லது.
இந்த query-ன் output பின்வருமாறு அமையும்.


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SELECT loan.date_lent, person.name, book.title
-> FROM loan, book, person
-> WHERE loan.book_id = book.book_id
-> AND loan.person_id = person.person_id;
date_lent	name	title
2012-11-19	Nithya	Sila nerangalil sila manithargal
2012-08-04	Shrinivasan	Tuesdays with Morrie
2012-12-12	Kanmani	Sila nerangalil sila manithargal
2012-12-31	Shrinivasan	Sila nerangalil sila manithargal
4 rows in set (0.00 sec)
mysql>

மேலும், நீங்கள் filters-ஐ இணைக்க விரும்பினால் AND keyword மூலம் நமக்கு வேண்டிய conditions-ஐ இணைத்து தேவைப்படும் வகையில் result-ஐ வரவைக்கலாம்.

SELECT loan.date_lent, person.name, book.title

FROM loan, book, person

WHERE loan.book_id = book.book_id

AND loan.person_id = person.person_id

AND person.name = 'Nithya';

nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SELECT loan.date_lent, person.name, book.title
-> FROM loan, book, person
-> WHERE loan.book_id = book.book_id
-> AND loan.person_id = person.person_id
-> AND person.name='Nithya';
+------------+--------+---------------------------------+
| date_lent | name | title |
+------------+--------+---------------------------------+
| 2012-11-19 | Nithya | Sila nerangalil sila manithargal |
+------------+--------+---------------------------------+
1 row in set (0.00 sec)
mysql>

10.3 Join keyword மூலம் tables-ஐ இணைத்தல்

Join keyword மூலம் tables இணைக்கப்படும்போது select statement-ன் from clause-ல் எந்தெந்த tables இணைக்கப்படவேண்டும் என்பதை Join எனும் keyword-ஐப் பயன்படுத்தி ஒன்றன்பின் ஒன்றாக குறிப்பிட வேண்டும். பின்னர் ON எனும் clause-ல் அந்த tables-க்கு இடையிலிருக்கும் primary key-foreign key தொடர்பைக் குறிப்பிட வேண்டும்.

இதற்கான syntax பின்வருமாறு.

SELECT columns

FROM table1

JOIN table2

ON table1.foreign_key = table2.primary_key;

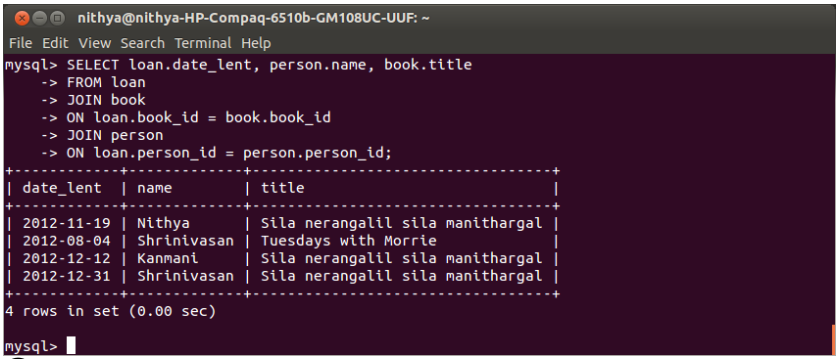
இவ்வாறு Join மூலம் tables-ஐ இணைத்து எழுதப்படும் queries, படிப்பதற்கு சுலபமாக இருக்கும். ஏனெனில் இந்த Joining conditions எல்லாம் ON clause-ல் எழுதப்படுவதால், இது where clause-ல் எழுதப்படும் conditions-லிருந்து வேறுபட்டுத் தெரிகிறது.

இந்த Join keyword மூலம் tables-ஐ நாம் Inner join மற்றும் Outer Join எனும் வெவ்வேறு வகைகளில் இணைக்க முடியும்.

10.4 Inner join மூலம் tables-ஐ இணைத்தல்

Where clause-ல் நாம் பயன்படுத்திய அதே உதாரணத்தை இங்கும் பயன்படுத்தி, அதனை Join keyword மூலம் எவ்வாறு இணைப்பது என்று பார்க்கலாம். இது பின்வருமாறு.

```
SELECT loan.date_lent, person.name, book.title

FROM loan

JOIN book

ON loan.book_id = book.book_id

JOIN person

ON loan.person_id = person.person_id;
```



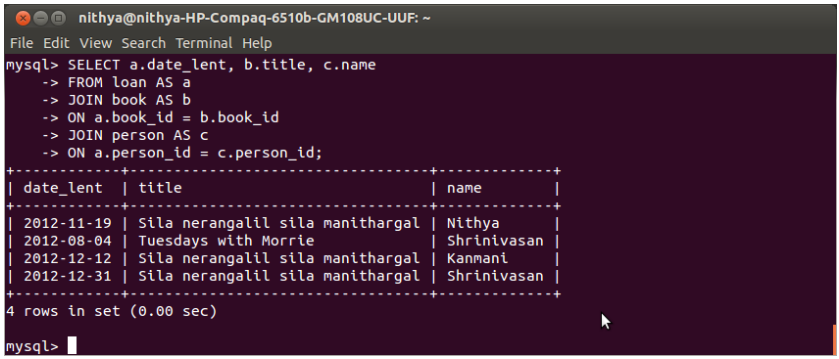
இந்த query-ல் tables அனைத்தும் Inner join மூலம் இணைக்கப்பட்டுள்ளன.

10.5 Table aliases-ஐப் பயன்படுத்துதல்

மேலே கொடுக்கப்பட்டுள்ள examples-ல் இருக்கும் columns அனைத்தும் அதன் முன்னர் table name-ஐ வைத்து குறிப்பிடப்படுகிறது. ஒருவேளை இந்த table nameமிகப்பெரிய பெயராக இருந்தால், ஒவ்வொரு முறையும் இந்தப் பெரிய பெயரைக் குறிப்பிட்டு அதன் column name-ஐக் குறிப்பிடுவது என்பது சற்று கடினமான விஷயம். எனவே அந்தப் பெரிய பெயருக்கு பதிலாக, ஒரு சிறிய பெயரை உருவாக்கி அதனை நாம் query-ல் பயன்படுத்தலாம். இதுவே alias name என்பதாகும்.

இந்த alias name எவ்வாறு உருவக்கப்படுகிறதெனில், clause-ல் table-ஐ குறிப்பிட்டவுடன் அதன் தொடர்ச்சியாக AS எனும் keyword-ஐப் பயன்படுத்தி ஒரு சிறிய பெயரைக் குறிப்பிடுவதன் மூலம் உருவாக்கப்படுகிறது. இதனை பின்வரும் உதாரணத்தில் காணலாம்.

```
SELECT a.date_lent, b.title, c.name
FROM loan AS a
JOIN book AS b
ON a.book_id = b.book_id
JOIN person AS c
ON a.person_id = c.person_id;
```



இங்கு a,b,c எனும் 3 alias name முறையே loan,book மற்றும் person எனும் 3 tables-க்கு உருவாக்கப்பட்டுள்ளது.

இந்த query-ல் நாம் முழு table name-க்கு பதிலாக alias-ஐ பயன்படுத்தத் தொடங்கியதும், அந்த query முழுவதும் alias-ஐ மட்டுமே பயன்படுத்த வேண்டும். இடையில் alias-க்கு பதிலாக மீண்டும் அந்த table-ன் முழு பெயரையும் பயன்படுத்த முடியாது.

10.6 Outer Join மூலம் tables-ஐ இணைத்தல்

Outer Join-ல் left outer join மற்றும் right outer join என இரு வகைகள் உண்டு.

இந்த Left outer join-ல் Join எனும் keyword-க்கு இடப்புறம் இருக்கும் table-ல் இருந்து அனைத்து row-வும் return செய்யப்படும். வலப்புறம் இருக்கும் table, Joining condition-ஐப் பொருத்து, இடப்புற table-ல் இருந்து வெளிப்படுத்தப்பட்ட மதிப்புகளுக்கு இணையான மதிப்பினைக் கண்டுபிடித்து வெளிப்படுத்தும். அப்படி இல்லையெனில் வலப்புற table-ல் இருந்து “NULL”மதிப்பு வெளிப்படுத்தப்படும்.

இதற்கான syntax பின்வருமாறு.

```
SELECT columns

FROM table1

LEFT OUTER JOIN table2

ON table1.foreign_key = table2.primary_key;
```

உதாரணத்துக்கு loan மற்றும் person எனும் table-லிலிருந்து , ஒவ்வொரு நபரும் சமீபத்தில் கடன் வாங்கிய தேதியைக் கண்டுபிடிக்க, Outer Join மூலம் query-யைப் பின்வருமாறு அமைக்கலாம்.

```
SELECT p.name, MAX(l.date_lent)

FROM person p

LEFT OUTER JOIN loan l

ON p.person_id = l.person_id

GROUP BY p.name;
```

இந்த Query-ன் output பின்வருமாறு அமையும்.

இதில் person எனும் table-ல் இருந்து அனைத்து நபர்களின் பெயர்களும் பட்டியலிடப்பட்டுவிட்டது. பின்னர் ஒவ்வொரு நபரும் கடன் வாங்கிய தேதி வலப்புற table-ல் இருந்து எடுக்கப்பட்டு வெளிப்படுத்தப்பட்டு விட்டது. இதில் கடனே வாங்காத ஒரு நபருக்கு வலப்புற table, “NULL”-எனும் மதிப்பினை வெளியிட்டிருப்பதைக் காணலாம்.

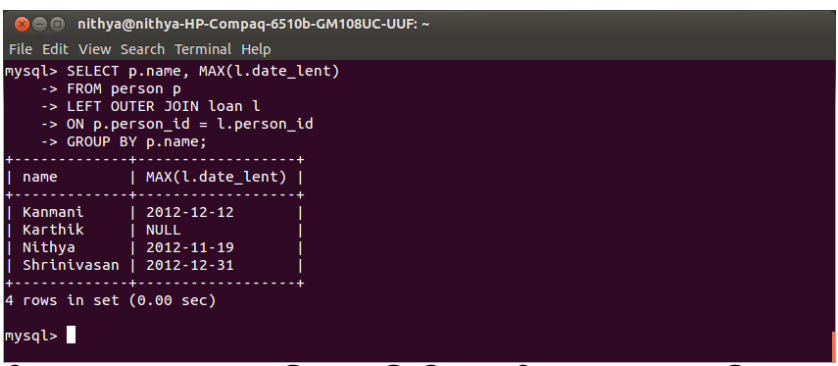
இவ்வாறே Right outer join-ம் வலப்புற table-ல் இருந்து அனைத்து மதிப்பினையும் வெளிப்படுத்திவிட்டு , அதற்கு இணையான மதிப்புகளை இடப்புற table-லிலிருந்து வெளிப்படுத்தும். அப்படி இணை மதிப்புகள் ஏதும் இடப்புற table-ல் இல்லையெனில் NULL மதிப்பினை வெளிப்படுத்தும்.

10.7 Column alias-ஐப் பயன்படுத்துதல்

Table-க்கு alias name-ஐ உருவாக்குவது போன்றே, Column-க்கும் alias name-ஐ உருவாக்கலாம். அப்படி உருவாக்க வேண்டிய தேவை என்னவெனில், ஒரு query-ன் select statement-ல் கொடுக்கப்படும் column-ன் மீது ஏதேனும் aggregate functions பயன்படுத்தப்பட்டால், அவை return செய்யும் மதிப்புகள், அந்த column name-ன் கீழ் வராது. எனவே இவ்வாறு return செய்யப்பட்ட மதிப்புகளின் தலைப்பில் என்ன column name இருக்க வேண்டும்

என்பதை அந்த column-க்கு ஒரு alias name-ஐ உருவாக்குவதன் மூலம் குறிப்பிடலாம். இது பின்வருமாறு.

```
SELECT p.name, MAX(l.date_lent) as maximum_date

FROM person p

LEFT OUTER JOIN loan l

ON p.person_id = l.person_id

GROUP BY p.name;
```



மேலும் ஒரு query இரண்டு வெவ்வேறு table-ல் இருந்து ஒரே பெயர் கொண்ட column-ல் இருக்கும் data-வை வெளிப்படுத்துகிறது எனில், நாம் result set-ஐப் பார்க்கும்போது, எந்த column எந்த table-லிலிருந்து எடுக்கப்பட்டது எனும் குழப்பத்தை விளைவிக்கும். எனவே இந்தக் குழப்பத்தைத் தவிர்ப்பதற்கு நாம் ஒரே பெயரில் இருக்கும் இரண்டு columns-க்கும் வெவ்வேறு alias name-ஐ உருவாக்குவதன் மூலம் வேறுபடுத்திக் காட்டலாம்.

10.8 Subquery மூலம் tables-ஐ இணைத்தல்

இரண்டு வெவ்வேறு queries ஒன்றுக்குள் ஒன்றாக வைக்கப்படுவதை subquery என்கிறோம். பொதுவாக select statement-ஐப் பயன்படுத்தும் ஒரு query-ல் சில conditions-ஐ வலியுறுத்த ஒரு paranthesis-இக்குள் கொடுக்கப்படும் மற்றொரு query அதன் subquery எனப்படும். முதலில் இந்த subquery-தான் execute செய்யப்படும். பின்னர் இந்த subquery-ஆல் கொடுக்கப்படும் மதிப்புகளை வைத்துதான் main query-ஆனது execute செய்யப்படும். இதற்கான syntax பின்வருமாறு.

SELECT columns

FROM table1

WHERE col1 IN (SELECT col2 FROM table2 WHERE …);

மேலே கொடுக்கப்பட்டுள்ள syntax-ல் subquery ஆனது where clause-ல் In operator-க்கு வலப்புறம் அமைந்துள்ளது. ஒரு subquery ஒரே ஒரு column-ஐத் தான் return செய்ய வேண்டும் என்பது விதி. எனவே இவ்வாறு return செய்யப்பட்ட ஒரு மதிப்பினை வைத்து main query-ஆனது ஒப்பீடு செய்யப்படுகிறது.

Subqueries-ஐ non-correlated subquery மற்றும் correlated subquery என இரண்டு வகையாகப் பிரிக்கலாம்.

10.9 Non-correlated subquery அமையும் விதம்

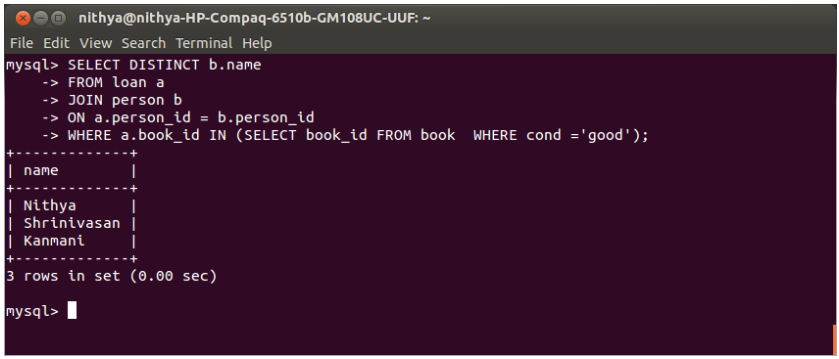
Non-Correlated subqueries-ல் வெளியே இருக்கும் outer query-

ல் உள்ள table-ஐயும் உள்ளே இருக்கும் subquery-ல் உள்ள table-ஐயும் இணைக்கும் வகையில் எந்த ஒரு condition-ம் இருக்காது.

உதாரணத்துக்கு ‘book’ எனும் table-ல் இருக்கும் ‘cond’ எனும் column ஒரு புத்தகம் எந்த நிலைமையில் உள்ளது என்பதை விளக்கும் வகையில் good அல்லது bad எனும் மதிப்புகளைப் பெற்றிருக்கிறது.இப்போது நாம் ‘cond’ எனும் column-ல் ‘good’ மதிப்பினைப் பெற்றிருக்கும் புத்தகங்களை மட்டும் எந்தெந்த நபருக்கு வழங்கியுள்ளோம் எனும் விவரத்தை non-correlated subqueries மூலம் சுலபமாக எடுக்கலாம். இது பின்வருமாறு.

```
SELECT DISTINCT b.name

FROM loan a

JOIN person b

ON a.person_id = b.person_id

WHERE a.book_id IN (SELECT book_id FROM book WHERE cond
='good');
```

இந்த query-ல் paranthesis-க்குள் கொடுக்கப்பட்ட subquery ஆனது முதலில் நல்ல நிலைமையில் இருக்கும் புத்தகங்களைப் பட்டியலிடும். பின்னர் மேலே கொடுக்கப்பட்டுள்ள main query–ஆனது இந்த மதிப்புகளை வைத்துக் கொண்டு, இத்தகைய புத்தகங்களை எந்தெந்த நபருக்கு வழங்கியுள்ளோம் எனும் விவரங்களைப் பட்டியலிடும். இதன் output பின்வருமாறு.



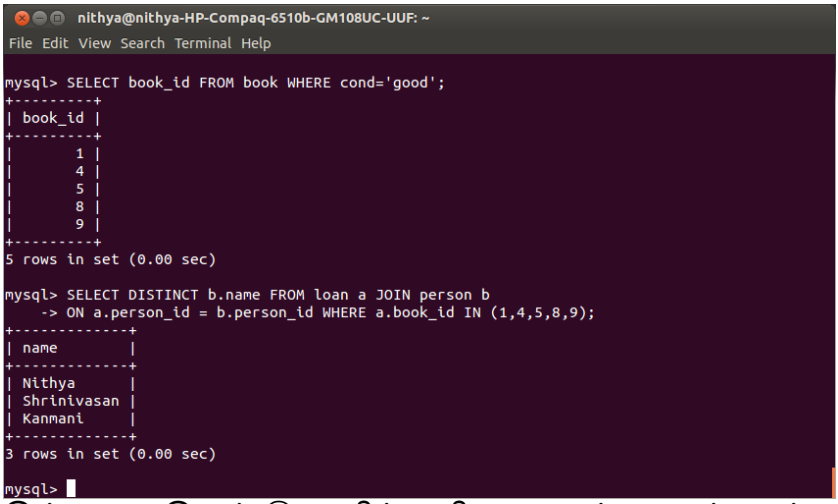
இந்த விவரங்களையே நீங்கள் subqueries-ஐப் பயன்படுத்தாமல் பெற விரும்பினால், தனித்தனியாக இரண்டு queries-ஐ execute செய்ய வேண்டியிருக்கும். இது பின்வருமாறு.

SELECT book_id FROM book WHERE cond = 'good';

SELECT DISTINCT b.name FROM loan a JOIN person b

ON a.person_id = b.person_id WHERE a.book_id IN **(1,4,5,8,9);**

முதல் query, எந்தெந்த புத்தகங்கள் நல்ல நிலைமையில் உள்ளன எனும் விவரத்தை வெளியிடும்.
பின்னர் இந்த result-ஐ condition-ல் வைத்துக்கொண்டு இரண்டாவது query, எந்தெந்த நபருக்கு இந்தப் புத்தகங்களை வழங்கியுள்ளோம் எனும் விவரத்தை வெளியிடும்.



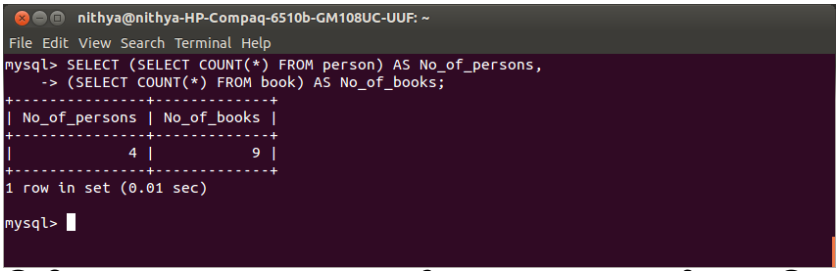
இவ்வாறு இரண்டு தனித்தனி queries-ன் மூலம் நாம் செய்யும் ஒரு வேலையை, subquery-யைப் பயன்படுத்தி ஒரே query-ல் செய்து விடலாம். இதுவே subquery-ன் ஒரு சிறந்த அம்சம் ஆகும்.

10.10 Select statement-ல் Subquery-ஐப் பயன்படுத்துதல்

Subquery-யை நாம் select statement-ல் பட்டியலிடப்படும் columns-க்குப் பதிலாகக்கூடப் பயன்படுத்தலாம். இதிலும் subquery-ஆனது ஒரே ஒரு column-ஐ மட்டுமே கொடுக்க வேண்டும். இல்லையெனில் MySQL, error-ஐ வெளிப்படுத்தும்.

பின்வரும் உதாரணத்தில் கொடுக்கப்பட்டுள்ள query, இரண்டு வெவ்வேறு table-களில் இருக்கும் data-வை summary செய்து தகவல்களை அனுப்பியிருப்பதைக் காணலாம்.
SELECT (SELECT COUNT(*) FROM person) AS No_of_persons,

(SELECT COUNT(*) FROM book) AS No_of_books;



இதில் select statement-ல் எப்போதும் பட்டியலிடப்படும் columns-க்குப் பதிலாக இங்கு இரண்டு வெவ்வேறு subqueries பட்டியலிடப்பட்டுள்ளது. இவை return செய்யும் மதிப்புகள் ஒரே row-வில் அமையப் பெற்றிருக்கும்.

10.11 Correlated subqueryஅமையும் விதம்

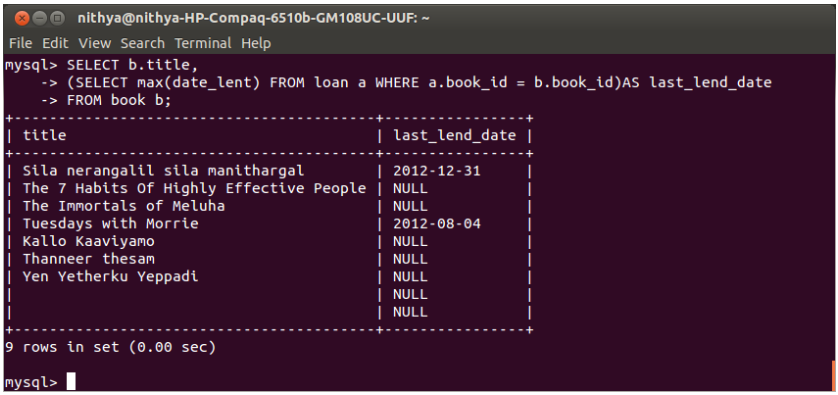
Correlated subqueries-ல் வெளியே இருக்கும் outer query-ல் உள்ள table-ம் உள்ளே இருக்கும் subquery-ல் உள்ள table-ம் ஒரு பொதுவான condition-ஆல் இணைக்கப்பட்டிருக்கும்.

உதாரணத்துக்கு book எனும் table-ல் இருந்து main query-ஆல் return செய்யப்படுகின்ற ஒவ்வொரு புத்தகத்துக்கும், அது ஒரு நபருக்கு எப்போது கடைசியாக வழங்கப்பட்டது எனும் தேதியைப் பட்டியலிட query-யைப் பின்வருமாறு அமைக்கலாம்.

SELECT b.title,

(SELECT max(date_lent) FROM loan a WHERE a.book_id = b.book_id)AS last_lend_date

FROM book b;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SELECT b.title,
-> (SELECT max(date_lent) FROM loan a WHERE a.book_id = b.book_id)AS last_lend_date
-> FROM book b;
title	last_lend_date
Sila nerangalil sila manithargal	2012-12-31
The 7 Habits Of Highly Effective People	NULL
The Immortals of Meluha	NULL
Tuesdays with Morrie	2012-08-04
Kallo Kaaviyamo	NULL
Thanneer thesam	NULL
Yen Yetherku Yeppadi	NULL
	NULL
	NULL
9 rows in set (0.00 sec)
mysql>

இதில் sub query-ல் இருக்கும் loan எனும் table-ம், main query-ல் இருக்கும் book எனும் table-ம், book_id எனும் ஒரு பொதுவான column-ஆல் இணைக்கப்பட்டுள்ளன. இதுவே correlated subquery என்பதாகும்.

ஏனெனில் subquery-ல் இருக்கும் ஒவ்வொரு row-வும் execute செய்யப்படுவதற்கு mainquery-ல் இருந்து ஒரு மதிப்பு தேவைப்படுகிறது. எனவே correlated subquery முறைப்படி ஒரு query-யை run செய்வதற்கு நீண்ட நேரம் பிடிக்கும்.

10.12 Union மூலம் queries-ஐ இணைத்தல்

UNION Operator என்பது இரண்டு அல்லது அதற்கு மேற்பட்ட queries மூலம் கிடைக்கும் விடைகளை ஒரே resultset-ஆக மாற்றப் பயன்படுகிறது. இதற்கான syntax பின்வறுமாரு.

SELECT columns FROM table1

UNION

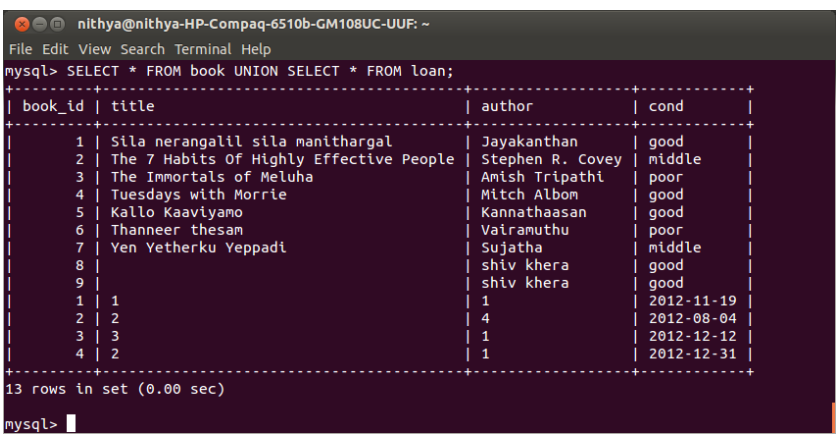
SELECT columns FROM table2;

உதாரணத்துக்கு பின்வரும் இரண்டு queries தனித்தனியாக execute செய்யப்பட்டாலும் அவை UNION Operator மூலம் இணைக்கப்பட்டிருப்பதால், அவை வெளிப்படுத்தும் result set ஒன்றாக அமையும்.

SELECT * FROM book

UNION

SELECT * FROM loan;



மேலும் Union Operator மூலம் இணைக்கப்படும் ஒவ்வொரு query-யிலும் ஒரே எண்ணிக்கையிலான columns இருக்க வேண்டும்.

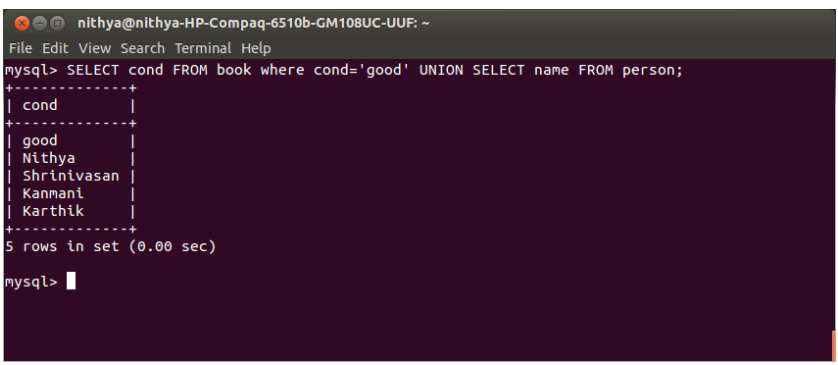
சாதாரணமாக இந்த Union Operator, ஒரே மாதிரியாக இருக்கும் மதிப்புகளை ஒரு முறைக்குமேல் மறுமுறை வெளிப்படுத்தாது. அவ்வாறு நீங்கள் வெளிப்படுத்த விரும்பினால், UNION என்பதற்கு பதிலாக UNION ALL எனும் keyword-ஐப் பயன்படுத்த வேண்டும்.

இந்த UNION மற்றும் UNION ALL-க்கான
வித்தியாசத்தினை பின்வரும் உதாரணத்தின் மூலம்
அறியலாம்.

SELECT cond FROM book where cond = 'good'

UNION

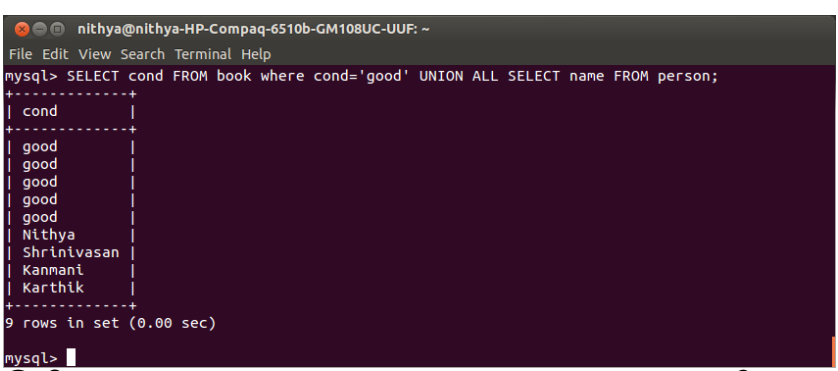
SELECT name FROM person;



SELECT cond FROM book where cond = 'good'

UNION ALL

SELECT name FROM person;



இதில் “good” எனும் வார்த்தை ஒரு தடவைக்குமேல்

பலமுறை வெளிப்பட்டிருப்பதைக் காணலாம்.

11

Library எனும் **database-**ல் சேமிக்கப்பட்டிருக்கும் **data-**ஐ மாற்றுதல்

காலத்தின் தேவைக்கேற்ப, நாம் பட்டியலில் சேமித்துவைத்திருக்கும் தகவல்களை மாற்றுவதற்கான சூழ்நிலை ஏற்படும். அப்போது சேமித்துவைக்கப்பட்டிருக்கும் தகவல்களை எவ்வாறு மாற்றுவது என்பது பற்றி இந்தப் பகுதியில் காணலாம்.

பொதுவாக 'update' எனும் command, table-ல் சேமிக்கப்பட்டிருக்கும் data-வை மாற்றப் பயன்படும். இதற்கான syntax பின்வருமாறு.

UPDATE table_name SET column_name = value WHERE condition;

இந்த syntax பின்வருமாறு விளக்கப்படுகிறது.

- UPDATE – இது எவ்வகையான query, run செய்யப்படுகிறது என்பதைக் குறிக்கும்.
- SET – இது மாற்றம் நிகழும் column-ன் பெயரையும் மற்றும் மாற்றப்பட்ட மதிப்பினையும் தாங்கியிருக்கும். ஒன்றுக்கு மேற்பட்ட column-ன் மதிப்புகள் மாற்றப்படவேண்டுமெனில், இதனை பின்வருமாறு அமைக்க வேண்டும்.

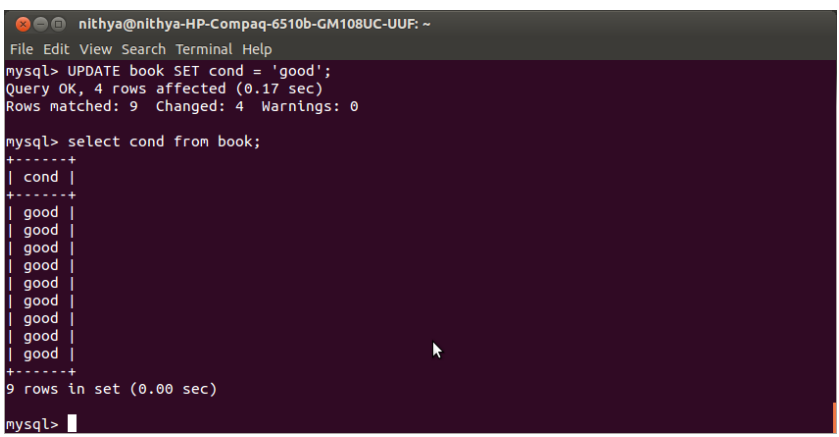
SET name0 = 'value', name1 = 'some_other_value',

…

- WHERE – இந்த clause எந்த ஒரு குறிப்பிட்ட row-வில் மதிப்புகள் மாற்றப்படவேண்டும் எனும் condition-ஐக் கொடுக்கப் பயன்படுகின்றன.

முதலில் ஒரு Column-ன் அனைத்து மதிப்புகளையும் எவ்வாறு மாற்றுவது என்று பார்க்கலாம். பின்வரும் எடுத்துக்காட்டில் இருக்கும் query, book எனும் table-ல் இருக்கும் 'cond' column-ன் அனைத்து மதிப்புகளையும் 'good' என்று மாற்றிவிடும்.

UPDATE book SET cond = 'good';

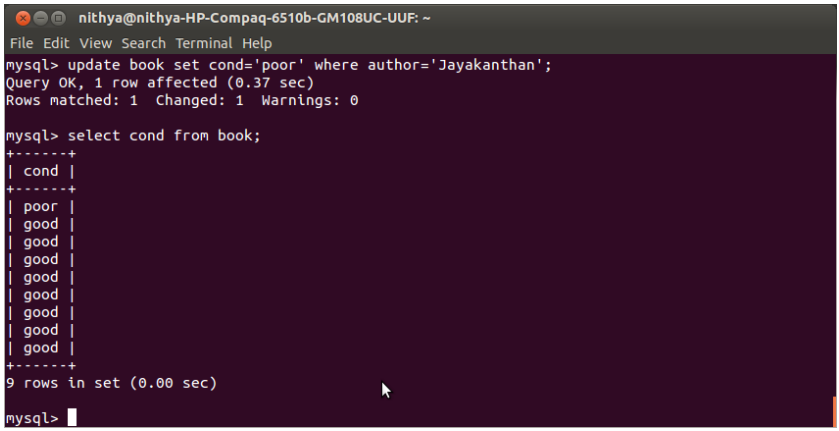


அடுத்ததாக ஒரே ஒரு row-வில் மட்டும் இந்த மாற்றம் நிகழ query-யைப் பின்வருமாறு அமைக்க வேண்டும்.

UPDATE book

SET cond = 'poor'

WHERE author = 'Jayakanthan';



இங்கு எந்த row-வில் இருக்கும் author-ன் பெயர் “Jayakanthan” என்று உள்ளதோ, அந்த row-க்கு மட்டும் ‘cond’ column, ‘poor’என்று மாற்றப்பட்டுள்ளது.

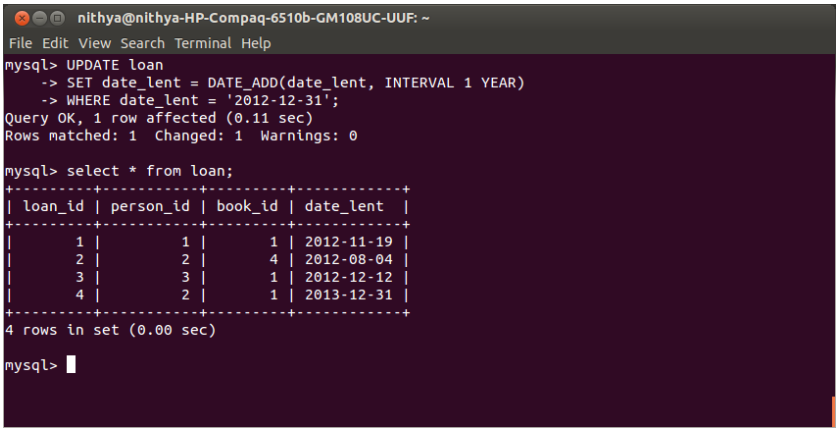
11.1 Function மூலம் column-ல் இருக்கும் மதிப்பினை மாற்றுதல்

பின்வரும் query-ல், loan எனும் table-ல் தவறுதலாக கொடுக்கப்பட்ட தேதியானது DATE_ADD function மூலமாக மாற்றப்படுவதைக் காணலாம். இந்த DATE_ADD function-ஆனது, ஏற்கனவே உள்ள தேதியுடன் ஒரு வருடம் கூடுதலாக இணைத்து, ஒரு புதிய தேதியாக மாற்றியுள்ளது.

UPDATE loan

SET date_lent = DATE_ADD(date_lent, INTERVAL 1 YEAR)

WHERE date_lent = ’2012-12-31’;



எனவே இது போன்ற update queries, table-ல் இருக்கும் மதிப்பினைக் கணக்கில் கொண்டே புதிய மதிப்புகளை உருவாக்குவதால் மிகவும் பயனுள்ளதாக அமையும்.

12

Library எனும் **database-**ல் சேமிக்கப்பட்டிருக்கும் **data-**ஐ நீக்குதல்

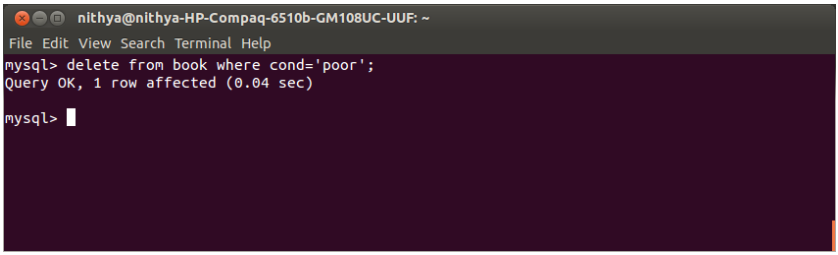
ஒரு table-ல் இருந்து ஒரு குறிப்பிட்ட row-வை மட்டும் அழிக்க ‘delete’ command பயன்படும். இதற்கான syntax பின்வருமாறு.

```
DELETE FROM table_name WHERE some_conditions;
```

- DELETE – இது தான் இந்த query-ன் துவக்கம். இந்த query அழிக்கும் வேலையைச் செய்யப் போகிறது என்பதைக்க் குறிக்கும்.
- FROM – எந்த table-ல் இருந்து நீக்க வேண்டும் என்பதைக் குறிக்கும்.
- WHERE – இங்கு conditions கொடுக்கப்பட்டு, அதில் பொருந்தும் rows அழிக்கப்படுகின்றன.

உதாரணத்துக்கு பின்வரும் query-ல் ‘book’ எனும் table-ல் இருந்து, ‘cond’ column-ல் poor எனும் மதிப்பினைக் கொண்ட row மட்டும் நீக்கப்பட்டுவிடும்.

```
DELETE FROM book WHERE cond = ‘poor’;
```


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> delete from book where cond='poor';
Query OK, 1 row affected (0.04 sec)
mysql>

12.1 பல்வேறு tables-ஐ இணைத்து rows-ஐ அழித்தல்

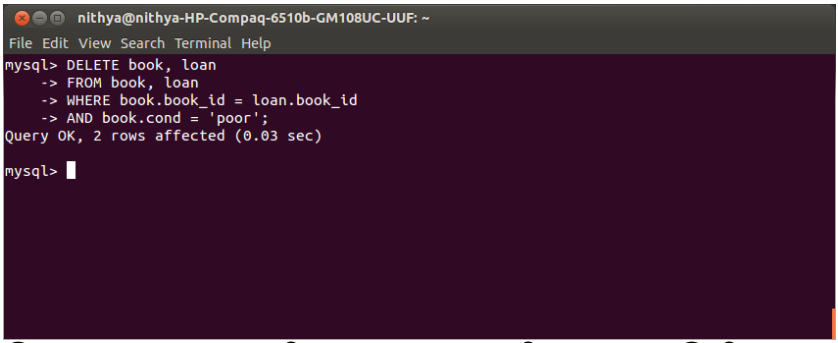
பின்வரும் query-ல் book மற்றும் loan எனும் இரண்டு tables-ல் உள்ள rows-ம் கொடுக்கப்பட்டுள்ள condition-ஐப் பொறுத்து அழிக்கப்படுவதைக் காணலாம்.

DELETE book, loan

FROM book, loan

WHERE book.book_id = loan.book_id

AND book.cond = ‘poor’;



இதற்கான syntax பின்வருமாறு விளக்கப்படுகிறது.

- Delete – இது தான் இந்த query-ன் துவக்கம். இந்த query அழிக்கும் வேலையைச் செய்யப் போகிறது என்பதைக்க் குறிக்கும்.
- list,of,tables – இங்கு தொடர்ச்சியாக கொடுக்கப்படும் tables-லிருந்து மட்டுமே rows அழிக்கப்படும்.
- FROM – எதிலிருந்து condition-ஐ எடுக்க வேண்டும் என்பதைக் குறிக்கும்.
- list,of,tables – இங்கு தொடர்ச்சியாக கொடுக்கப்படும் tables மட்டுமே where clause-ல் condition-ஐ வலியுறுத்தப் பயன்படும்.
- WHERE – இங்கு conditions கொடுக்கப்பட்டு, அதில் பொருந்தும் rows அழிக்கப்படுகின்றன.

12.2 தற்காப்பு நடவடிக்கைகள்

MySQL-ல் செய்தவற்றை திரும்ப எடுத்துக்கொள்ளும் ‘undo’ எனும் அமைப்பு கிடையாது. நீங்கள் ஒருமுறை,

DROP DATABASE library;

என்பது போன்ற ஒரு query-யை run செய்துவிட்டால்,

‘library’ எனும் database-ல் இருக்கும் அனைத்து tables-ம் அழிக்கப்பட்டுவிடும்.

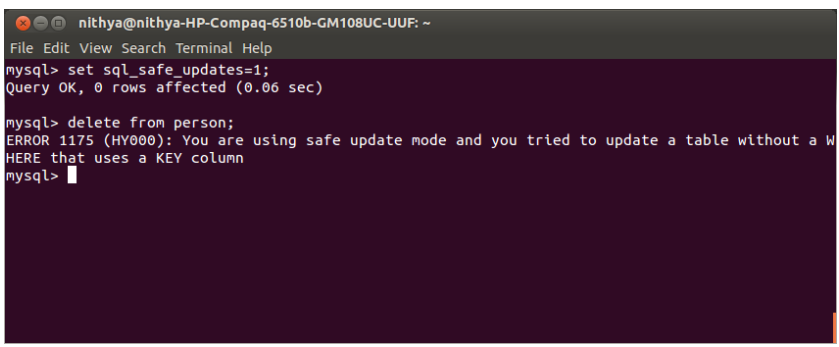
தவறுதலாக இவ்வகையான queries , run செய்யப்பட்டது என்றால், அதனை undo செய்யமுடியாது.
வேண்டுமானால் logs, backups அல்லது data recovery tool மூலமாக அழிக்கப்பட்ட தகவல்களைத் திரும்பப் பெற முடியும்.

எனவே பாதுகாப்பான முறையில் ‘library’ database-ல் இருக்கும் data-வை நீக்குவதற்கு பின்வரும் குறிப்புகளைப் பின்பற்றலாம்.

- Users-க்கு, தேவைக்கு அதிகமான permissions-ஐ கொடுக்கக்கூடாது. உதாரணத்துக்கு ஒரு database-லிருந்து நீங்கள் தகவல்களைப் பெற மட்டுமே விரும்பினால், அந்த database-ன் root account-ஐப் பயன்படுத்தத் தேவையில்லை. மாறாக, வெறும் select queries மட்டும் run செய்ய permissions இருக்கும் ஒரு user-ஐ create செய்துவிட்டு, பின்னர் அந்த user மூலமாக login செய்யலாம். அவ்வாறு செய்யும்போது, நாம் வெறும் adhoc queries-ஐ run செய்து வேண்டிய தகவல்களைப் பெற மட்டுமே உரிமை கிடைத்திருக்கும்.
- Server Administrator என்பவரால், MySQL-ல் சேமித்து வைக்கப்பட்டிருக்கும் அனைத்து தகவல்களும் தினந்தோறும் backup எடுத்து வைக்கப்பட்டிருக்க வேண்டும்.
- தேவையானபோது adhoc backups-ம் எடுத்து வைத்துக் கொள்ள வேண்டும். இந்த adhoc backups என்பது server admin என்பவர் backup எடுக்கத் தவறினாலும், நாமே நமக்குத் தேவையான வகையில் எடுத்து வைத்துக்கொள்ளும் backup ஆகும்.
- பின்வரும் option, delete மற்றும் update போன்ற queries-ஐ LIMIT அல்லது WHERE clauses இல்லாமல் நேரடியாக இயக்கப்படுவதிலிருந்து தடுக்கிறது.

ஏனெனில் இந்த queries நேரடியாக இயக்கப்படும்போது (conditions இல்லாமல்) அந்த முழு table-ம் மாற்றப்படும் அல்லது அழிக்கப்படும் அபாயம் உள்ளது. எனவே இதுபோன்ற options-ஐப் பயன்படுத்துவதன் மூலம் இதுபோன்ற அபாயங்கள் நிகழாமல் தடுக்கலாம்.

SET SQL_SAFE_UPDATES = 1;



- MyISAM அல்லது வேறு ஏதாவது non-transactional storage engine-ல் வேலைசெய்யும்போது , உங்களுடைய queries-ஐ உண்மையான table-ல் run செய்யாமல் அதைப்போன்றே பிரதியெடுக்கப்பட்ட ஒரு table-ல் run செய்து பரிசோதித்துப் பார்க்கலாம்.

உதாரணத்துக்கு பின்வரும் command, book எனும் table-ஐ copy செய்து test எனும் ஒரு table-ஐ உருவாக்கும்.

CREATE TEMPORARY TABLE test LIKE book;

INSERT test SELECT * FROM book;

பின்னர் நமது commands-ஐ எல்லாம் முதலில் test எனும் table-ல் run செய்து பரிசோதித்துவிட்டபின் book எனும் table-ல் சென்று run செய்யலாம்.

ஆனால் இதுபோன்ற அணுகுமுறை பெரிய அளவில் தகவல்களை சேமித்து வைத்திருக்கும் databases-க்குப் பொருந்தாது.

- InnoDB அல்லது BerkeleyDB போன்ற storage engine-ல் எவ்வாறு queries-ஐ run செய்வது என்பதைப் பின்வருமாறு பார்க்கலாம்.ஒரு transaction-க்குள் கொடுக்கும் queries ஒழுங்காக வேலை செய்கிறது என உறுதி செய்யப்பட்டபின் அந்த மாற்றத்தை COMMIT செய்து நிலைப்படுத்தலாம்.

  அப்படி இல்லையெனில் ROLLBACK செய்து அந்த மாற்றம் நிகழாமல் அதன் பழைய நிலைக்கே கொண்டு வந்துவிடலாம்.

  உதாரணத்துக்கு பின்வருமாறு கொடுக்கப்பட்டிருக்கும் இரண்டு transaction-ல், முதலில் rollback-ம் அடுத்ததாக commit-ம் செய்யப்படுகிறது.

```
BEGIN;

UPDATE book SET author = “Sujatha”;

SELECT * FROM book;

ROLLBACK;

BEGIN;

UPDATE book SET AUTHOR = “Sujatha”

WHERE author = “Vairamuthu”;
```

```
SELECT * FROM book;

COMMIT;
```

13

Users-ஐ கையாளுதல்

MySQL என்பது பலப்பல users பயன்படுத்தக்கூடிய database ஆகும். இந்தப் பாகத்தில் எவ்வாறு அத்தகைய பயனர்களை கண்காணிப்பது மற்றும் அவர்களுக்குத் தேவையான அனுமதியை வழங்குவது என்பதைப் பற்றிப் பார்ப்போம்.
MySQL's privilege system என்பது ஒவ்வொரு பயனரும் அவர்களுக்கு அனுமதிக்கப்பட்டுள்ள வகையில் மட்டுமே database -ஐ பயன்படுத்த முடியும் என்பதை விளக்குகிறது. ஒவ்வொரு பயனரும் ஒரு username மற்றும் அந்த database இணைப்பின் remote address கொண்டு அடையாளம் கண்டுகொள்ளப்படுகிறார். பின்னர் அந்த பயனர், அவருடைய சரியான password-ஐ கொடுப்பதன் மூலம் அந்த database -வுடன் இணைக்கப்படுகிறார்.

13.1 Server-ஐ பயன்படுத்த ஒரு புதிய user-ஐ உருவாக்குதல்

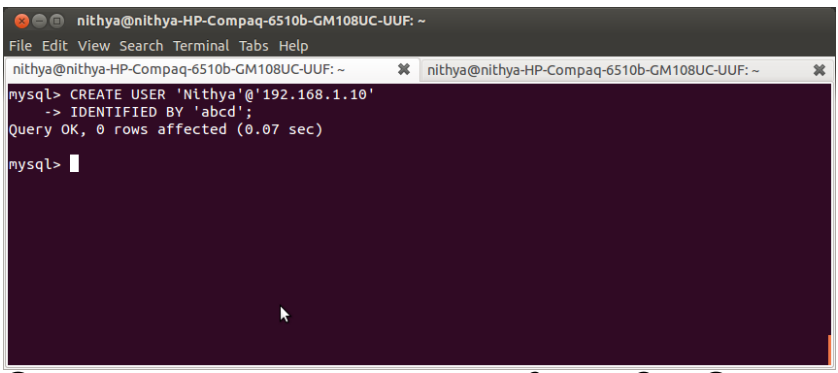
ஒரு புதிய user-ஐ உருவாக்குவதற்கு, நீங்கள் admin rights கொண்டவராக இருக்க வேண்டும். பொதுவாக, MySQL-ஐ install செய்யும்போது உருவாக்கப்படும் 'root' user, இத்தகைய 'admin rights' கொண்ட user-ஆக அமையும். இதற்கான syntax பின்வருமாறு.

CREATE USER user@host IDENTIFIED BY 'password';
பின்வரும் எடுத்துக்காட்டில், 'Nithya' எனப்படும் ஒரு புதிய user, 192.168.1.10 எனும் remote IP address-க்கு உருவாக்கப்படுகிறார். இவர் இந்த remote IP address-ல் மட்டுமே 'abcd' எனும் password-ஐ பயன்படுத்தி login செய்யமுடியும். வேறு ஏதாவது ip address-ல் password சரியாக கொடுக்கப்பட்டாலும் கூட இவரால் login செய்ய

முடியாது.

CREATE USER ‘Nithya’@’192.168.1.10′

IDENTIFIED BY ‘abcd’;

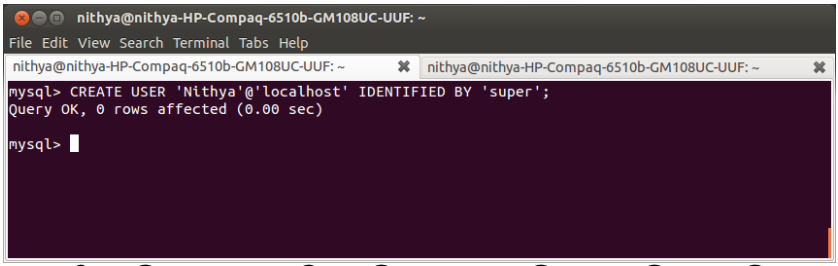


இங்கு பயனர்களை உருவாக்கும்போது கொடுக்கும் முகவரி, ip address ஆகவோ, local host name ஆகவோ அல்லது முழுவதுமாக கொடுக்கப்படும் domain name ஆகவோ இருக்கலாம்.

தற்போது நமது கணினியில் ஏற்கனவே இயங்கிக் கொண்டிருக்கும் MySQL database-ஐக் குறிப்பிட localhost என்பதைப் பயன்படுத்தலாம். இது பின்வருமாறு.

CREATE USER ‘Nithya’@’localhost’

IDENTIFIED BY ‘super’;



எனவே இதுவரை கொடுக்கப்பட்டுள்ள இரண்டு எடுத்துக்காட்டிலும் 'Nithya' எனும் ஒரே user உருவாக்கப்பட்டாலும், அவர் வெவ்வேறு password-ஐ பயன்படுத்தி வெவ்வேறு முகவரிகளில் நுழைய அனுமதி பெறுகிறார்.
இந்த create user எனும் command, MySQL version 5.0.2 மற்றும் அதற்கு மெற்பட்ட version-ல் மட்டுமே காணப்படும். இதற்கு முந்தைய version-களில் இந்த command கிடையாது. மேலும் பழைய version-களில் database அனுமதிகள் பற்றிய விவரங்களெல்லாம், MySQL tables-ஆன user,host மற்றும் database-ல் காணப்படும். எனவே பழைய version-களில் ஒரு புதிய பயனரை உருவாக்க, அதற்குத் தேவையான தகவல்களையெல்லாம் user table-ல் insert command-ஐப் பயன்படுத்தி செலுத்துவதன் மூலம் பெறலாம். இது பின்வருமாறு.

INSERT INTO user (Host, User, Password)

VALUES ('123.456.78.90', 'Nithya',

PASSWORD('abcd'));

FLUSH PRIVILEGES;

இங்கு கொடுக்கப்பட்டுள்ள query-ல் 'abcd'எனும் password மதிப்பு மட்டும், PASSWORD() function மூலம் கொடுக்கப்படுகிறது. ஏனெனில் இவ்வாறு கொடுப்பதன் மூலம் மட்டுமே இந்த password, encrypt செய்யப்பட்ட எழுத்துக்களாக இந்த user table-ல் பதிவு செய்யப்படும்.
மேலும் இந்த FLUSH PREVILEGES எனும் command மிக முக்கியமான ஒன்றாகும். இது ஒவ்வொரு முறை user table-ல் data மாற்றப்படும்போதும், privileges-ஐ reload செய்யப் பயன்படுகிறது. இதனை create user command-வுடன் பயன்படுத்தத் தேவையில்லை.

13.2 ஏற்கனவே உள்ள user-ஐ server-ல் இருந்து நீக்குதல்

DROP USER எனும் command-ஆனது, ஒரு user-வுடைய தகவல்களை 'user' table-லிலிருந்து நீக்கப் பயன்படுகிறது.

DROP USER Nithya@localhost;

ஒரு user நீக்கப்பட்டுவிட்ட பின்னர், அவர் தற்போது நிலுவையில் பயன்படுத்திக்கொண்டிருக்கும் active user session-லிருந்து உடனடியாக நீக்கப்படமட்டார். ஆனால் மேலும் பல புதிய session-களை உருவாக்குவதிலிருந்து தடை செய்யப்படுவார்.

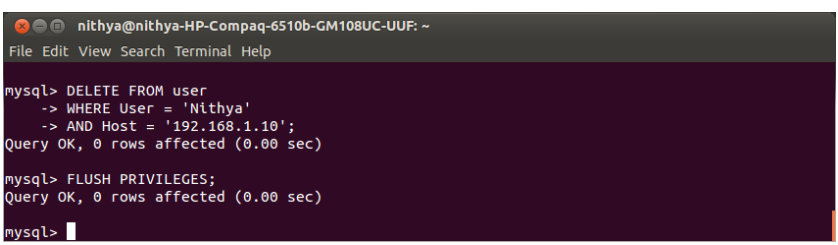
இந்த drop user எனும் command, MySQL version = > 5.0.2-வில் மட்டுமே காணப்படும். அதற்கு முந்தைய version-களில் ஒரு பயனரை நீக்குவதற்கு delete commandதான் பயன்படும் . இது பின்வருமாறு.

DELETE FROM user

WHERE User = ‘Nithya’

AND Host = ’192.168.1.10’;

FLUSH PRIVILEGES;



13.3 ஒரு user-க்கு பெயர் மாற்றம் செய்தல்

இது மிக அரிதாக நடைபெறும் செயல் ஆகும். ஒரு username எப்போது மாற்றப்படுமெனில், ஒரு சில அலுவலகங்களில் ஒரு நபருக்கு கொடுக்கப்படும் username அவருடைய surname-ஐப் பொருத்து அமையும். அப்போது அந்த நபரின் திருமணத்துக்குப் பின்

அவரின் surname மாற்றப்படுவதால், அவருடைய username-ம் அதைப்பொருத்து மாற்றப்பட வேண்டியிருக்கும்.

எனவே “நித்யா துரைசாமி” என்று இருக்கும் ஒருவருடைய பெயர், திருமணத்துக்குப் பின் “நித்யா சீனிவாசன்” என மாற்றப்படுகிறது எனில், RENAME USER command -ஐப் பயன்படுத்தி, இதனை பின்வருமாறு மாற்றலாம்.

RENAME USER old_user@host TO new_user@host;
ஆனால் இந்த rename user எனும் command, MySQL version >= 5.0.2-வில் மட்டுமே காணப்படும். அதற்கு முந்தைய version-களில் ஒரு பயனருக்கு பெயர் மாற்றம் செய்ய update commandதான் பயன்படும் . இது பின்வருமாறு.

UPDATE user

SET User = ‘Nithya_Shrinivasan’

WHERE User = ‘Nithya_Duraisamy’

AND Host = ‘localhost’;

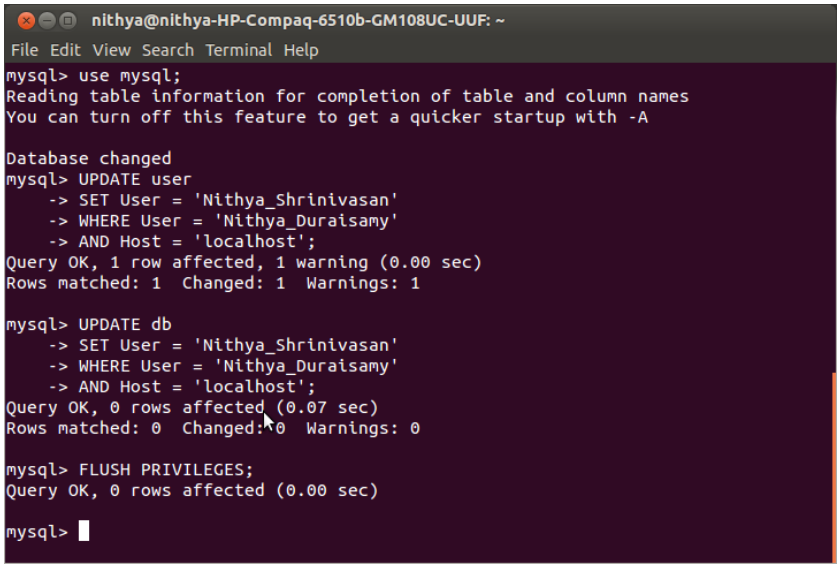
UPDATE db

SET User = ‘Nithya_Shrinivasan’

WHERE User = ‘Nithya_Duraisamy’

AND Host = ‘localhost’;

FLUSH PRIVILEGES;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> UPDATE user
-> SET User = 'Nithya_Shrinivasan'
-> WHERE User = 'Nithya_Duraisamy'
-> AND Host = 'localhost';
Query OK, 1 row affected, 1 warning (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 1
mysql> UPDATE db
-> SET User = 'Nithya_Shrinivasan'
-> WHERE User = 'Nithya_Duraisamy'
-> AND Host = 'localhost';
Query OK, 0 rows affected (0.07 sec)
Rows matched: 0 Changed: 0 Warnings: 0
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
mysql>

இந்த update command, முதலில் user table-ஐயும், பின்னர் அதன் தொடர்பான rows-ஐ database table-லிலும் மாற்றும்.

13.4 Wildcards-மூலம் பல்வேறு IP address-க்கு ஒரே முறையில் users-ஐ உருவாக்குதல்

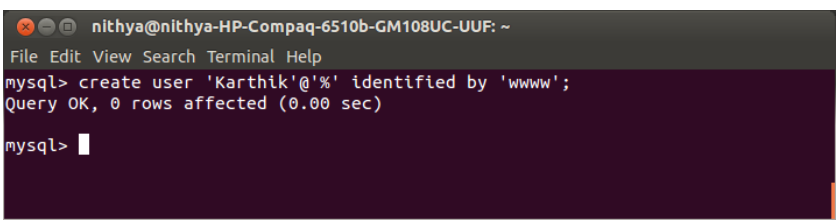
Percent (%) மற்றும் Underscore(_)எனும் இரண்டு wildcard characters-ஐப் பயன்படுத்தி நாம் ஒரே மாதிரியான pattern-ல் இருக்கும் ip address-ஐக் கண்டுபிடித்து, பின்னர் கண்டுபிடிக்கப்பட்ட அனைத்து IP முகவரிகளுக்கும் பயனர்களை உருவாக்கலாம். இது பின்வருமாறு

CREATE USER ‘Karthik’@’192.168.0.%’

IDENTIFIED BY ‘abcd’;

இதில் 192.168.0.x எனும் IP range-ல் துவங்கும் அனைத்து IPமுகவரிகளுக்கும் பயனர்கள் உருவாக்கப்பட்டுள்ளனர்.
மேலும் ஒரு குறிப்பிட்ட முகவரியில் மட்டும் அல்லாமல், அனைத்து முகவரிகளிலும் login செய்வதற்குத் தேவையான username மற்றும் password-ஐ உருவாக்க query-யை பின்வருமாறு அமைக்கலாம்.

CREATE USER ‘Karthik’@’%’ IDENTIFIED BY ‘phrasebook’;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> create user 'Karthik'@'%' identified by 'wwww';
Query OK, 0 rows affected (0.00 sec)
mysql>

13.5 ஒரு database/table-க்கு அனுமதி பெற்ற பயனர்களைப் பட்டியலிடல்

SELECT Db FROM db

WHERE User = ‘user’ AND Host = ‘host’;

SELECT Table_name FROM tables_priv

WHERE User = ‘user’

AND Host = ‘host’ AND Db = ‘db’;
ஒரு பயனர் எந்தெந்த database-ஐப் பயன்படுத்துகிறார் என்பதைக் கண்டறிய ‘db’ table-ஐ user மற்றும் host-ஐப்

பயன்படுத்தி query- இடுவதன் மூலம் கண்டறியலாம்.

SELECT Db, Table_name FROM tables_priv

WHERE User = ‘Nithya’ AND Host = ‘localhost’

மேலும் அந்த user, எந்தெந்த tables-ஐப் பயன்படுத்துகிறார் என்பதைக் கண்டறிய tables_priv எனும் table-ஐ user, host மற்றும் db போன்ற விவரங்களை condition-ல் கொடுத்து query-இடுவதன் மூலம் காணலாம்.

13.6 Password -ஐ மாற்றுதல்

நீங்கள் MySQL -இல் ஒரு சாதாரண user-ஆக login செய்யும்போது, உங்களுடைய password-ஐ மாற்றி அமைக்க query-யை பின்வருமாறு அமைக்கலாம்.

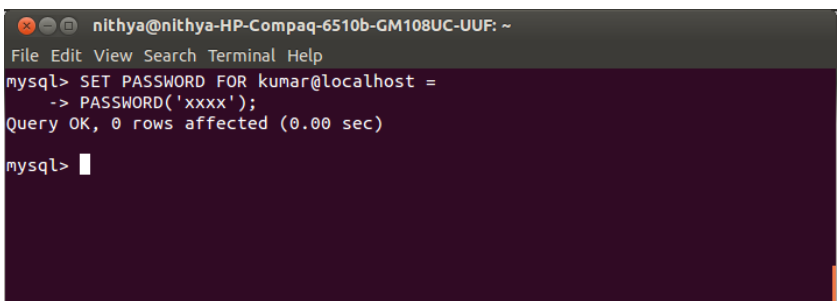
SET PASSWORD = PASSWORD(‘murali’);

இங்கு ‘murali’ என்பது நீங்கள் புதிதாக மாற்றியுள்ள உங்களுடைய password ஆகும்.

மேலும் நீங்கள் ஒரு admin-ஆக login செய்யும்போது, மற்றவர்களுடைய password-ஐ மாற்றி அமைக்க பின்வரும் query-யைப் பயன்படுத்தலாம்.

SET PASSWORD FOR kumar@localhost =

PASSWORD('xxxx');


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> SET PASSWORD FOR kumar@localhost =
-> PASSWORD('xxxx');
Query OK, 0 rows affected (0.00 sec)
mysql>

இங்கு 'xxxx' என்பது நீங்கள் 'kumar'-க்கு புதிதாக மாற்றியுள்ள password ஆகும்.
மேலும் பின்வரும் update query -கூட இதே வேலையைச் செய்கிறது.

UPDATE user

SET Password = PASSWORD('xxxx')

WHERE User = 'kumar' AND Host = 'localhost';

இவ்வாறு நீங்கள் set password command-ஐப் பயன்படுத்தும்போது மட்டும் FLUSH PRIVILEGES -ஐப் பயன்படுத்தத் தேவையில்லை என்பதை நினைவில் கொள்க.

13.7 Users-க்கு நமது tables-ஐப் பயன்படுத்துவதற்கான privileges-ஐ வழங்குதல்

Grant command-ஐப் பயன்படுத்தி ஒரு database அல்லது tables-ஐ அணுகுவதற்குத் தேவையான அனுமதிகளை நாம் பிற users-க்கு வழங்கலாம். இதற்கான syntax பின்வருமாறு.

GRANT privileges ON db.table

TO user@host

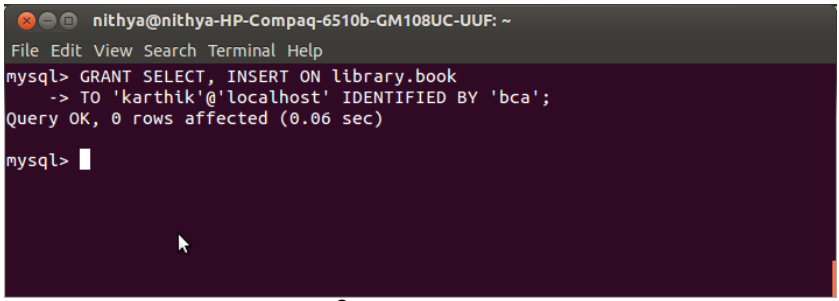
IDENTIFIED BY ‘password’;

உண்மையில் சொல்லபோனால், MySQL version 4.1-க்கு முன்னர் இந்த GRANT command- ஐத் தான் ஒரு புதிய user-ஐ உருவாக்கப் பயன்படும்.
நாம் ஒரு user-க்கு வேண்டிய அனைத்து அனுமதிகளையும் ஒரு comma மூலம் பிரித்து ஒரே வரியில் கொடுத்துவிடலாம்.

பின்வரும் எடுத்துக்காட்டில் karthik-எனும் user-க்கு library database-ல் உள்ள ‘book’ table-ஐ select செய்யவும் insert செய்யவும் மட்டுமே அனுமதிகள் வழங்கப்பட்டுள்ளன. இவருக்கு data-வை மாற்றவோ அழிக்கவோ அனுமதி கிடையாது.

GRANT SELECT, INSERT ON library.book

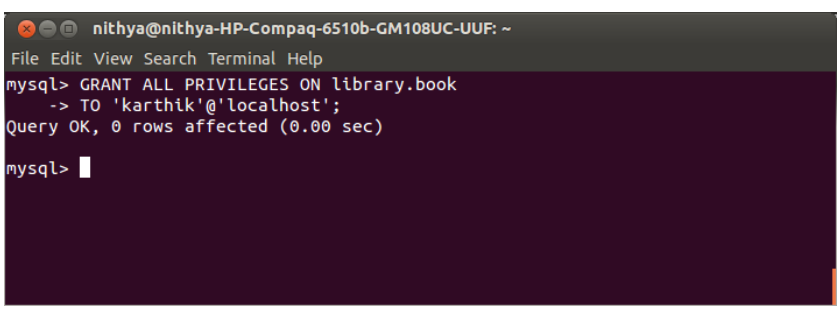
TO ‘karthik’@’localhost’ IDENTIFIED BY ‘bca’;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> GRANT SELECT, INSERT ON library.book
-> TO 'karthik'@'localhost' IDENTIFIED BY 'bca';
Query OK, 0 rows affected (0.06 sec)
mysql>

ஒரு user-க்கு பட்டியலில் உள்ள அனைத்து அனுமதிகளையும் வழங்க, “ALL PRIVILEGES” அல்லது வெறும் “ALL” எனும் keyword-ஐப் பயன்படுத்தலாம். பின்வரும் உதாரணத்தில் ‘kumar’ எனும் user-க்கு library database-ல் உள்ள ‘book’ table-ஐ அணுகுவதற்கான அனைத்து அனுமதிகளும் வழங்கப்பட்டுள்ளன.

GRANT ALL PRIVILEGES ON library.book

TO ‘karthik’@’localhost’;


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> GRANT ALL PRIVILEGES ON library.book
-> TO 'karthik'@'localhost';
Query OK, 0 rows affected (0.00 sec)
mysql>

நீங்கள் ஏற்கனவே நிலுவையில் இருக்கும் ஒரு user-க்கு அனுமதிகளை வழங்கினால், அப்போது IDENTIFIED BY clause -ஐப் பயன்படுத்த தேவையில்லை.

13.8 ஒரு User தனக்கிருக்கும் privileges

முழுவதையும் மற்றவருக்கு வழங்குதல்

ஒரு User அனைத்து privileges -ஐயும் பெற்றிருந்தாலும் கூட அவரால் தன்னுடைய privileges முழுவதையும் மற்றவருக்கு வழங்க முடியாது.

இவ்வாறு அவர் வழங்க விரும்பினால், முதன்முதலில் அவருக்கு அனுமதிகள் வழங்கப்படும்போதே with grant option – என்பதுடன் சேர்த்து வழங்கப்பட்டிருக்க வேண்டும். அப்போதுதான் அவரால் தன்னுடைய privileges-ஐ மற்றவருக்கு வழங்க முடியும். இதற்கான syntax பின்வருமாறு.

GRANT privileges ON db.table

TO user@host

IDENTIFIED BY ‘password’

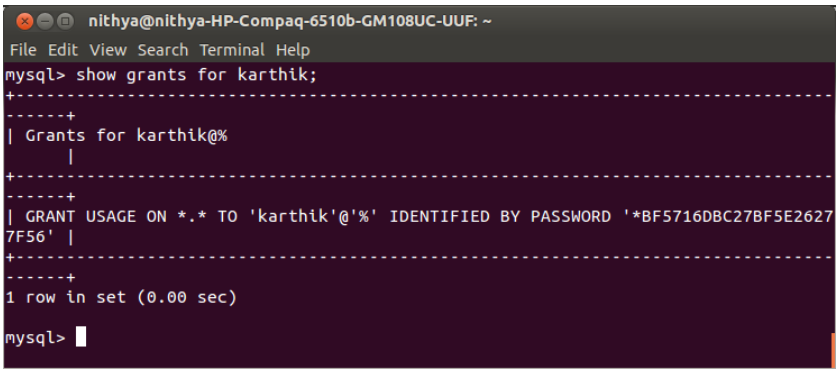
WITH GRANT OPTION;

மேலும், ஒருவரால் தனக்கு இல்லாத privileges-ஐ மற்றவருக்கு கொடுக்க முடியாவிட்டாலும்கூட, வெவ்வேறு permissions கொண்ட இரண்டு users தங்களுடைய privileges -ஐப் பகிர்ந்து கொள்ளலாம்.

13.9 ஒரு User-க்கு இருக்கும் privileges-ஐக் காணுதல்

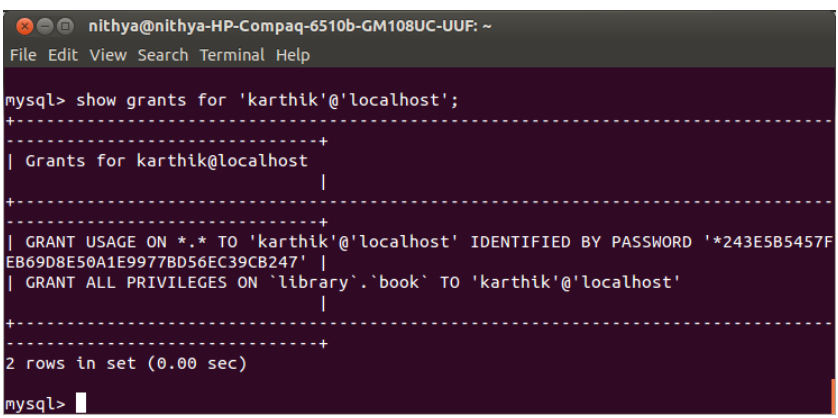
Show grants எனும் command-ஐப் பயன்படுத்தி ஒரு user-க்கு இருக்கும் அனைத்து privileges-ஐயும் காணலாம்.

SHOW GRANTS FOR karthik;



மேலும் பின்வரும் command-ன் மூலம், நாம் அவருக்கு அனுமதிகளை கொடுக்கப் பயன்படுத்தியிருக்கும் Grant statements-ஐப் பட்டியலிடலாம்.

SHOW GRANTS FOR ‘karthik’@’localhost’;



இவ்வகையான statement-ல் இடம்பெறும் password, நீங்கள் உண்மையான password-ஐ அறிந்து கொள்ளாத

வகையில் encrypt செய்யப்பட்டதாக இருக்கும்.

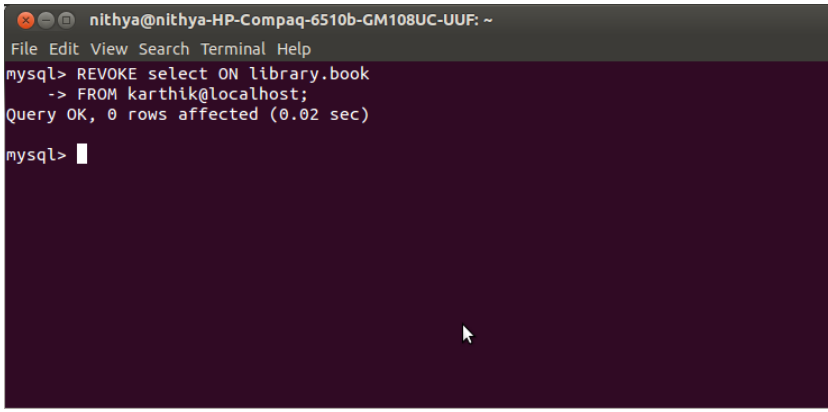
FOR clause இல்லாமல் கொடுக்கப்படும் grant statement ஆனது user-ஐ current_user() – ஆக கணக்கில்கொண்டு, அவருக்குரிய privileges-ஐக் காட்டும்.

13.10 ஒரு user-க்கான அனுமதிகளை நீக்குதல்

Revoke-ஐப் பயன்படுத்தி ஒரு பயனருக்கு உண்டான privileges-ஐ நீக்கலாம். என்னென்ன privileges நீக்கப்படவேண்டும் என்பதை வரிசையாக இடையில் comma-வைக் கொடுத்தோ, அல்லது ALL எனும் keyword-ஐப் பயன்படுத்தியோ அனைத்து privileges-ஐயும் நீக்கிவிடலாம்.

பின்வரும் எடுத்துக்காட்டில் உள்ள query, ‘karthik’ எனும் user-க்கு select privilege-ஐ மட்டும் library database-ல் உள்ள book table-லிலிருந்து நீக்கிவிடுகிறது.

```
REVOKE select ON library.book

FROM karthik@localhost;
```


nithya@nithya-HP-Compaq-6510b-GM108UC-UUF: ~
File Edit View Search Terminal Help
mysql> REVOKE select ON library.book
-> FROM karthik@localhost;
Query OK, 0 rows affected (0.02 sec)
mysql>

பின்வரும் எடுத்துக்காட்டில் உள்ள query, ஒரு user-க்கு உண்டான grant option-வுடன் சேர்த்து அனைத்து privileges-ஐயும் நீக்கிவிடுகிறது.

REVOKE ALL PRIVILEGES, GRANT OPTION

ON sampdb.mytable

FROM ‘shrini’@’localhost’;

13.11 Network access-ஐ செயலிழக்க செய்தல்

TCP/IP மூலம் வரும் remote access-ஐ செயலிழக்க செய்ய my.cnf-வுடன் skip-networking-ஐ இணைக்கவும் அல்லது –skip-networking switch ஐப் பயன்படுத்தி mysqld-ஐ start செய்யவும். பின்னர் இந்த MySQL, localhost-லிருந்து socket வழியாக வரும் இணைப்பை மட்டுமே பெற்றுக்கொள்ளும்.

```
# mysqld_safe --skip-networking &
```

Network access-ஆனது disable செய்யப்பட்டுவிட்டதெனில், user-ஆல் remote hosts வழியாக MySQL-வுடன் இணைப்பை ஏற்படுத்த முடியாது.

நீங்கள் MySQL-வுடன் remote access-ஐ அனுமதித்தாலும், MySQL-க்கும் வெளிப்புற உலகுக்கும் இடையிலிருக்கும் firewall, அதற்குரிய appropriate port-ல் இணைப்பை உண்டாக்கும். பொதுவாக, 3306 எனும் port இதற்காகப் பயன்படுத்தப்படும்.

13.12 User authentication-ஐ செயலிழக்க வைத்தல்

சிலசமயங்களில் நீங்கள் உங்களுடைய password-ஐ மறந்துவிட நேரிடும். அப்போது உங்களுடைய புதிய password-ஐ reset செய்ய, இந்த MySQL-ஐ password கேட்காத வகையில் restart செய்ய வேண்டும்
அவ்வாறு செய்வதற்கு நீங்கள் Admin rights பெற்றவராக இருக்க வேண்டும். முதலில் MySQL server-ஐ நிறுத்தி விட்டபின், உங்களுடைய master my.cnf file-வுடன் skip-grant-tables என்பதை இணைத்து பின் restart செய்யவும். மாறாக – - skip-grant-tables எனும் switch-வுடன் சேர்த்தும் mysqld-ஐ restart செய்யலாம்.

```
# mysqld_safe --skip-grant-tables &
```

இவ்வாறு restart செய்வதன்மூலம், நாம் password இல்லாமலேயே பின்வருமாறு இந்த MySQL-வுடன் இணைக்கப்பட முடியும்.

MySQL -u root
இவ்வாறு password இல்லாமல் நாம் MySQL-வுடன் இணைக்கப்பட்டுவிட்ட பின்னர், எப்போதும்போல் நாம் நம்முடைய புதிய password-ஐ அமைத்துக் கொள்ளலாம்.

```
update user set password=PASSWORD("NEW-ROOT-
PASSWORD") where User='root';
```

மீண்டும் user authentication-ஐ enable செய்ய FLUSH PRIVILEGES command-ஐப் பயன்படுத்தலாம் அல்லது MySQL server-ஐ --skip-grant-tables இல்லாமல் restart செய்யலாம்.

13.13 SSL -மூலம் பாதுகாப்பான இணைப்பை உருவாக்குதல்

MySQL அதன் database server-க்கும் client-க்கும் இடையில் encrypt செய்யப்பட்ட network இணைப்பை ஏற்படுத்துவதற்கு SSL-ஐப் பயன்படுத்தும்.
இந்த அமைப்பு, உங்கள் server-வுடைய compile நேரத்திலேயே enable செய்யப்பட வேண்டும். மேலும்இது enable செய்யப்பட்டுள்ளதா என்பதை சரிபார்க்க பின்வரும் command-ஐப் பயன்படுத்தலாம்.

SHOW VARIABLES LIKE 'have_openssl';
இந்த command, "Yes" அல்லது "No" என்பதை அதன் variable-வுடன் சேர்த்து வெளிப்படுத்தும்.
ஒருவேளை இந்த அமைப்பு compile time-ல் enable செய்யப்படவில்லையெனில், நீங்கள் மீண்டும் உங்கள் server-ஐ –with-vio மற்றும் –with-openssl போன்ற configure switches-ஐப் பயன்படுத்தி recompile செய்ய வேண்டும்.

ஒரு encrypt செய்யப்பட்ட SSL இணைப்பை ஏற்படுத்துவதற்கு, மூன்று வகையான certificate files-ஐ வைத்திருக்க வேண்டும் அல்லது அவற்றை உருவாக்க வேண்டும். அவை certificate authority certificate, server certificate மற்றும் client certificate ஆகும்.

முதலில் my.cnf file-ல் [mysqld] பகுதியின் கீழ், பின்வரும் வரிகளை இணைக்கவும்.

இவை இந்த மூன்று வகையான certificate files வைக்கப்பட்டுள்ள இடத்தைக் குறிப்பிடும்.

ssl-ca = /path/to/CA-cert

ssl-cert = /path/to/server-cert

ssl-key = /path/to/client-cert

பின்னர் அதே file-ல் [client] பகுதியின் கீழ் server settings-ஐப் போன்றே client settings-ஐயும் இணைக்கவும். எனவே நீங்கள் server-க்குப் பயன்படுத்தும், அதே மூன்று certificate files-ஐயே client-க்கும் பயன்படுத்தலாம்.

மாறாக, நீங்கள் ஒவ்வொரு certificate-ஐயும் MySQL-க்கு command-line argument-ஆக பின்வருமாறு குறிப்பிடலாம்.

MySQL –ssl-ca = /path/to/CA-cert \

–ssl-cert = /path/to/server-cert \

–ssl-key = /path/to/client-cert

14

MySQL-ன் Application Programming Interface - API

MySQL server-ஐ ஒரு பயனர் சுலபமான வகையில் தொடர்பு கொள்வதற்காக உருவாக்கப்பட்ட ஒன்றே API ஆகும். பல்வேறு programming மொழிகளில் எழுதப்படுகின்ற இந்த API-ல் ஏதேனும் ஒன்றை நாம் தேர்ந்தெடுத்துப் பயன்படுத்தலாம். ஒரு குறிப்பிட்ட மொழியில் API இல்லையென்றாலும், Open database connectivity எனப்படுகின்ற ODBC துணையுடன் நாம் MySQL server-ஐ தொடர்பு கொள்ள முடியும்.

14.1 C மொழியில் இயங்கும் API

இந்த C மொழியில் இருக்கும் API-தான், மற்ற மொழிகளில் இயங்கும் API-க்கு அடித்தளமாக விளங்குகிறது. அதாவது மற்ற மொழிகளின் API-ல் இருக்கும் ஒவ்வொரு function call-ம் , இந்த C library function-வுடன் தான் நேரடியாக தொடர்பு கொண்டிருக்கும்.

இந்த C API-ஐ பயன்படுத்த நீங்கள் உங்களுடைய program-ஐ libmysqlclient.so-வுடன் இணைக்க வேண்டும். இந்த library-ஆனது source-லிருந்து MySQL-ஐ compile செய்யும்போது தானாகவே install செய்யப்படும். வெவ்வேறு components-ஆக பிரிக்கப்படுகின்ற binary distributions-க்காக, MySQL-devel-* என்னும் பெயர் கொண்ட ஒன்றை நமது platform-ல் install செய்ய வேண்டும். இது libraries மற்றும் header files package-ஐ உள்ளடக்கியிருக்கும்.

MySQL API-ஐ பயன்படுத்தும் ஒரு C program, கண்டிப்பாக MySQL.h எனும் file-ஐ உள்ளடக்கியிருக்க வேண்டும். இந்த file உங்கள் system-ல் ஏற்கனவே இருக்கும் include directory-ல் காணப்படும். இது பின்வருமாறு.

#include <mysql/mysql.h>

libmysqlclient.so-வுடன் இணைப்பது என்பது compiler-ஐப் பொறுத்து மாறுபடும். பொதுவாக gcc மூலம் எவ்வாறு mytest.c எனும் பெயர் கொண்ட ஒரு program, compile செய்யப்பட்டு libmysqlclient-வுடன் இணைக்கப்படுகிறது என்பதை பின்வரும் எடுத்துக்காட்டில் காணலாம்.

shell> gcc o mytest mytest.c -lmysqlclient

இவ்வாறு compile செய்யும்போது, ஏதேனும் “error messages” வந்தால், mysql_config எனும் utility- யைப் பயன்படுத்தி இன்னும் என்னென்ன compiler options தேவை என்பதைக் கண்டுபிடிக்கலாம். மேலும் –cflags, –include மற்றும் –libs போன்ற switches-வுடன் சேர்த்து mysql_config-ஐ run செய்யும்போது, இது இன்னும் சில அதிகமான தகவல்களைக் கொடுக்கும். இது பின்வருமாறு.

shell> mysql_config –cflags –include –libs

-I/usr/include/mysql -mcpu=i486 -fno-strength-reduce

-I/usr/include/mysql

-L/usr/lib/MySQL -lMySQLclient lcrypt -lnsl -lm -lz -lc -lnss_files -lnss_dns -lresolv

-lc -lnss_files -lnss_dns lresolv

14.2 MySQL-வுடன் இணைதல்

MySQL database-வுடன் இணைப்பதற்கு mysql_init() மற்றும் mysql_real_connect() எனப்படுகின்ற இரண்டு function calls தேவை.

முதல் function, MySQL type-ஐச் சேர்ந்த ஒரு object-ஐ initialize செய்யும். இரண்டாவது function, இந்த object-ஐ அதன் argument-ஆக பயன்படுத்திக்கொள்ளும். பின்னர், அடுத்தடுத்து இருக்கும் API calls-ல் , இந்த object எந்த database connection-ஐ பயன்படுத்த வேண்டும் என்பதை வலியுறுத்த உதவும் “Resource argument”-ஆகப் பயன்படுத்தப்படும்.

Port argument என்பது database-வுடன் இணைப்பதற்கு உதவும் TCP/IP Port-ஆக பயன்படுத்தப்படும். இதன் மதிப்பு localhost இணைப்பிற்கு 0-ஆக இருக்கும்.

MYSQL mysql;
mysql_init(&mysql);
mysql_real_connect(&mysql, “host”, “user”,
“password”, “dbname”,
port, unix_socket, client_flag);
நீங்கள் இணைப்பிற்காக, வேறு எதாவது socket-ஐயோ அல்லது named pipe-ஐயோ பயன்படுத்தாதவரை unix socket argument-ன் மதிப்பு NULL-ஐப் பெற்றிருக்கும். அதேபோல் mysql client library-ன் ஒரு சில features-ஐ enable செய்யாதவரை, client_flag argument-ன் மதிப்பும் NULL-ஆகவே இருக்கும்.

இணைப்பு வெற்றிகரமாக நிகழ்ந்துவிட்டதெனில் mysql_real_connect() -ஆல் return செய்யப்படும் மதிப்பு ஒரு mysql object ஆகும். இந்த object-ம் முதலில் argument-ஆக அனுப்பப்பட்ட object-ம் ஒன்றே ஆகும். எனவே இதனை ஒரு புதிய variable-க்கு assign செய்யத் தேவையில்லை. மேலும், இணைப்பு ஏதேனும் ஒரு காரணத்தால்வெற்றிகரமாக நிகழ வில்லையெனில், இந்த function, “NULL” மதிப்பினை return செய்யும்.

14.3 Query-யை execute செய்தல்

mysql_query() function ஒரு query-யை execute செய்ய உதவும். இந்த function-க்கு argument-ஆக அனுப்பப்படும் query ஒரு “NULL terminated string”-ஆக இருக்க வேண்டும். மேலும் இறுதியில் கொடுக்கப்படும் semicolon தேவையில்லை. இந்த query, mysql object-ஐப் பயன்படுத்தி உருவாக்கப்பட்ட database connection-ல் execute செய்யப்படும்.

ஒரு query-யானது, binary data-வை உள்ளடக்கியிருக்கிறது என்றால், நாம் mysql_real_query() எனும் function-ஐப் பயன்படுத்தி அந்த query-ன் length-ஐக் குறிப்பிட வேண்டும்.
ஏனெனில் binary data-ஆனது “” எனும் character-ஐக் கொண்டிருக்கிறதெனில், இதனை அந்த mysql_query() function, string-ன் இறுதி எழுத்து குறிப்பிடப்படுவதற்கான அறிகுறியாக இந்த “”-ஐக் கணக்கில் கொள்ளும். எனவே binary data-வைக் கொண்டுள்ள query-க்கு mysql_real_query() function-ஐப் பயன்படுத்துவது சிறந்தது.

mysql_query(&mysql, “query”);

mysql_real_query(&mysql, “query”, length);
Query-யானது வெற்றிகரமாக execute செய்யப்பட்டுவிட்டதெனில், இந்த query function கொடுக்கும் மதிப்பு “NULL”-ஐப் பெற்றிருக்கும். அப்படி இல்லாமல் ஏதேனும் ஒரு தவறு எற்பட்டுவிட்டதெனில், இந்த function திருப்பி அனுப்பும் மதிப்பு, பின்வரும் அட்டவணையில் இருக்கும் மதிப்புகளில் ஏதேனும் ஒன்றாக இருக்கும்.

Error Codes from mysql_query()

CR_COMMANDS_OUT_OF_SYNC – Commands were executed in an improper order.

CR_SERVER_GONE_ERROR – The MySQL server has gone away.

CR_SERVER_LOST – The connection to the server was lost during the query.

CR_UNKNOWN_ERROR – An unknown error occurred.

14.4 Result set-லிருந்து தகவல்களைப் பெறுதல்

நீங்கள் ஒரு query மூலம் return செய்யப்படும் தகவல்களை எடுப்பதற்கு முன்னால், அந்த query-ன் result-ஐ MYSQL_RES எனும் object-க்கு assign செய்ய வேண்டும். mysql_use_result() எனும் function கடைசியாக execute செய்யப்பட்ட query-ன் result-ஐ இத்தகைய object-க்கு assign செய்ய உதவும். இவ்வாறு result-ஐ object-க்கு assign செய்துவிட்டபின், இந்த reference-ஐ அழிக்காமல் நாம் அடுத்தடுத்த query-ஐ execute செய்யலாம். வெற்றிகரமாக query-ஆனது execute செய்யப்பட்டால், இந்த function, “NULL” மதிப்பை திருப்பி அனுப்பும். அப்படி இல்லையெனில், முன்னர் குறிப்பிட்ட அட்டவணையில் இருக்கும் ஏதேனும் ஒரு error code-ஐ திருப்பி அனுப்பும்.

```
MYSQL_RES result;

MYSQL_ROW row;

result = mysql_use_result(&mysql);

row = mysql_fetch_row(result);
```

mysql_fetch_row() எனும் function, execute செய்யப்பட்ட query-லிருந்து தகவல்களை ஒவ்வொரு row-வாக எடுத்து அதனை mysql_row structure-ல் வெளிப்படுத்தும். அவற்றை நாம் row[0]-வில் தொடங்கி, row[n-1]முடிய ஒவ்வொரு row-வாக அணுகலாம். இதில் n என்பது அந்த data set-ல் இருக்கும் column-ன் எண்ணிக்கையைக்

குறிப்பிடும்.

mysql_num_fields() என்பது column-ன் எண்ணிக்கையையும், அவ்வாறே mysql_num_rows() என்பது rows-ன் எண்ணிக்கையையும் கொடுக்கப் பயன்படும்.

14.5 Error Messages-ஐ வெளிப்படுத்துதல்

mysql_errno() மற்றும் mysql_error() எனும் functions தவறுகள் ஏற்படும்போது, அந்த தவறுக்கு உண்டான என்னையும் மற்றும் அது எவ்வகையான தவறு என்பதை விளக்கும் ஒரு message-ஐயும் வெளிப்படுத்தும்.

இந்த function பயன்படுத்தும் arguments என்பது MySQL இணைப்பின் object தான். மேலும் இந்த function-ஆல் return செய்யப்படுகின்ற error பற்றிய தகவல்களெல்லாம், இந்த இணைப்பில் கடைசியாக execute செய்யப்பட்ட query-ல் இருக்கும் error பற்றிய தகவல்களே ஆகும். எனவே கடைசியாக execute செய்யப்பட்ட query வெற்றிகரமாக செய்யப்பட்டதெனில் mysql_errno() எனும் function, 0-வையும் மற்றும் mysql_error() எனும் function, NULL மதிப்பையும் return செய்யும்.

mysql_errno(&mysql);

mysql_error(&mysql);
நீங்கள் இவ்வகையான functions-ஐ ஒரு database-வுடன் இணைக்கும்போதோ அல்லது ஒரு query-யை execute செய்யும்போதோ எவ்வகையான தவறுகள் ஏற்பட்டுள்ளன என்பதைக் கண்டுபிடிக்கப் பயன்படுத்தலாம்.

14.6 இணைப்பை நிறுத்துதல்

MySQL இணைப்பைத் தொடங்கி, அந்த இணைப்பில்

நமக்குத் தேவையான வேலைகளைச் செய்து முடித்துவிட்ட பின்னர், அந்த இணைப்பை நிறுத்துவதற்கு mysql_close() எனும் function பயன்படும். இது mysql_init() மூலம் allocate செய்யப்பட்ட resource-ஐ de-allocate செய்யப் பயன்படும்.

```
mysql_close(&mysql);
```

உதாரணம்

```
#include <stdio.h>

#include <mysql/mysql.h>

main() {

MYSQL mysql;

MYSQL_RES *result;

MYSQL_ROW row;

int numrows, numcols, c;

mysql_init(&mysql);

/* Establish a database connection */

if (!mysql_real_connect(&mysql, “localhost”,

“username”, “password”,

“dbname”, 0, NULL, 0))

{

fprintf(stderr,

“Failed to connect to database: Error %d:

%s\n”, mysql_errno(&mysql),
```

```
mysql_error(&mysql));
}
/* Execute a query */
char query[] = “SELECT book_id, cond, title
FROM book”;
if (mysql_query(&mysql, query))
{
fprintf(stderr,
“Error executing query: Error %d: %s\n”,
mysql_errno(&mysql), mysql_error(&mysql));
}
/* Assign the result handle */
result = mysql_use_result(&mysql);
if (!result)
{
fprintf(stderr,
“Error executing query: Error %d: %s\n”,
mysql_errno(&mysql), mysql_error(&mysql));
}
/* Find the number of columns in the result */
numcols = mysql_num_fields(result);
/* Loop through the result set to display it */
```

```
while (row = mysql_fetch_row(result)) {

for(c=0; c<numcols; c++) {

printf("%s\t", row[c]);

}

printf("\n");

}

}
```

14.7 Perl மொழியில் இயங்கும் API

DBI (Database Interface) மற்றும் DBD (Database Driver) மூலம் perl மொழியில் MySQL-ஐ தொடர்பு கொள்ள முடியும். இதற்கு Perl 5.6.0 மற்றும் அதற்கும் மேற்பட்ட versions தேவை.
DBI ஏற்கனவே install செய்யப்படவில்லையெனில் Cpan மூலம் இதனை download செய்து install செய்யலாம். இது பின்வருமாறு.

```
shell> cpan

cpan> install DBI
```

மேலும் windows platform-ல் Activeperl distribution-ஐப் பயன்படுத்தி, ppm.bat script-ன் மூலம் perl modules-ஐப் பின்வருமாறு install செய்யலாம்.

```
C:\perl\bin> ppm.bat

ppm> install DBI
```

அவ்வாறே MySQL database driver-ஐ இணைக்க, DBD::MySQL module-ஐ அதே முறையில் install செய்யவும்.

cpan > install DBD::mysql

MySQL DBD-யைப் பயன்படுத்தும் perl script-ஆனது, பின்வரும் வரியை உள்ளடக்கியிருக்க வேண்டும்.

use Mysql;

14.8 MySQL-வுடன் தொடர்பு கொள்ளுதல்

MySQL object-ன் மேல் செயல்படும் connect method மூலம் நாம் database-வுடன் இணைக்கலாம். இது dbh எனப்படும் database handle-ஐக் கொடுக்கும். மேலும் localhost-வுடன் தொடர்பை ஏற்படுத்துவதற்கு undef என்பதை முதல் argument-ஆகப் பயன்படுத்தலாம். ஒரே ஒரு DBI method, C API-ல் உள்ள mysql_init() மற்றும் mysql_real_connect() என்ற இரண்டு functions-ஐயும் அழைத்துவிடும்.

$dbh = Mysql->connect(host, dbname,

user, password);

14.9 Query-யை execute செய்தல்

ஒரு query-யை execute செய்ய, dbh-ன் மேல் sql statement-ஐ argument-ஆக வைத்து , query() method-ஐ அழைக்கவும். இது sth எனப்படும் statement handle-ஐக் கொடுக்கும்.

$sth = $dbh->query(query);

14.10 Result set-லிருந்து தகவல்களைப் பெறுதல்

@row = $sth->fetchrow;

sth-ன் மேல் செயல்படும் fetchrow() method-ஆனது array அமைப்பில் data-வை வெளிப்படுத்தும். முதல்முறை இந்த fetchrow method அழைக்கப்படும்போது, அது dataset-ல் இருக்கும் முதல் row-வை வெளிப்படுத்தும். அவ்வாறே அதன் அடுத்தடுத்த row-வை வெளிப்படுத்த அழைப்புகளை அனுப்பும். கடைசியில் வெளிப்படுத்த records ஏதும் இல்லாத நிலையில், NULL-ஐ வெளிப்படுத்தும்.
இவ்வாறே sth-ன் மேல் செயல்படும் numrows எனும் method, rows-ன் எண்ணிக்கையையும், numfields எனும் method, columns-ன் எண்ணிக்கையையும் வெளிப்படுத்தும்.

14.11 Error Messages-ஐ வெளிப்படுத்துதல்

errno மற்றும் errstr போன்ற methods முறையே error எண்ணையும் மற்றும் error பற்றிய தகவலையும் தெரிவிக்க உதவுகின்றன. இந்த methods ஆனது, dbh-ன் மீது அழைக்கப்படும்போது, அது கடைசியாக execute செய்யப்பட்ட query-ல் இருக்கும் தவறுகளைப் பற்றிய விவரங்களைத் தருகிறது.

$errno = $dbh->errno;

$errstr = $dbh->errstr;

அவ்வாறே ஒரு connection-ஐ உருவாக்கும்போது உண்டாகும் தவறுகளைப் பற்றி தெரிந்து கொள்ள, அதே methods-ஐ MySQL object-ன் மீது அழைக்கவும்.

$errno = Mysql->errno;

$errstr = MySQL->errstr;

14.12 இணைப்பை நிறுத்துதல்

இணைப்பை நிறுத்துவதற்கென எந்த ஒரு தனி DBI method-ம் கிடையாது. ஒரு program-ஐ exit செய்யும்போது, resources எல்லாம் தானாகவே விடுவிக்கப்பட்டுவிடும். Exit செய்வதற்கு முன்னால், resource-ஐ விடுவிக்க விரும்பினால் undef எனும் command-ஐ handle-ன் மீது பயன்படுத்திச் செய்யலாம்.

உதாரணம்

#!/usr/bin/perl

use Mysql;

/* Establish a database connection */

$dbh = Mysql->connect(undef, “dbname”,”username”, “password”)

or die (“Failed to connect to database: Error ”

. Mysql->errstr);

/* Execute a query */

$sql_statement = “SELECT book_id, cond, title FROM book”;

```
$sth = $dbh->query($sql_statement)

or die (“Error executing query: Error ” .

$dbh->errno);

/* Loop through the result set to display it */

while (@row = $sth->fetchrow) {

for($i=0; $i<$sth->numfields; $i++) {

print $row[$i] . “\t”;

}

print “\n”;

}
```

14.13 PHP மொழியில் இயங்கும் API

PHP Applications-ல் பெரும்பாலானவை MySQL database-ஐத்தான் பயன்படுத்துகின்றன.

PHP enable செய்யப்பட்ட web server-ல் MySQL-ன் support-ஐப் பற்றித் தெரிந்து கொள்ள <?php phpinfo();?> என்பதை உள்ளடக்கிய ஓர் எளிய script-ஐ உருவாக்கவும்.

இந்த script-ஐ web browser-ல் திறந்து MySQL support அல்லது MySQLi support எனும் பகுதியைப் பார்க்கவும். MySQLi அதாவது MySQL Improved என்பது PHP API-ன் ஒரு புதிய version ஆகும். இது MySQL 4.1.3 மற்றும் அதற்கு மேற்பட்ட version-களில் இயங்கக்கூடியது. இதனை procedural முறையிலோ அல்லது object-oriented முறையிலோ பயன்படுத்தலாம். இந்தப் பகுதியில் MySQLi-ஐப் பற்றிக் காண்போம். பழைய MySQL API என்பது, இந்த MySQLi-ன் procedural பயன்பாட்டிலேயே அடங்கிவிடும். Compile time-ல் MySQLi support-ஐ –with-

mysql=/path/to/mysql_config எனும் switch-ஐப் பயன்படுத்தி enable செய்யலாம்.

14.14 MySQL-வுடன் இணைத்தல்

$conn = mysqli_connect(“host”, “user”,

“password”, “dbname”);

$conn = new mysqli(“host”,

“user”, “password”,”dbname”);
மேலே உள்ள முதல் syntax-ல் mysqli_connect() functions கொடுக்கப்பட்டுள்ள arguments-ஐப் பயன்படுத்தி ஒரு dbh அதாவது database connection handle-ஐக் கொடுக்கிறது. இரண்டாவதாக இருக்கும் syntax-ல் இதே வேலை, MySQLi object-ன் மீது constructor method-ஐப் பயன்படுத்துவதன் மூலம் செய்யப்படுகிறது.
இவ்வாறு procedural முறையிலோ அல்லது object-oriented முறையிலோ இணைப்பை உருவாக்கலாம். இதன் பின்னால் தொடரும் database operations எல்லாம் mysqli_connect()-லிருந்து arguments-ஐப் பயன்படுத்தி, விடைகளை அனுப்பும் functions-ஆகவோ அல்லது MySQLi object-ன் ஒரு புதிய instance மீது அழைக்கப்படும் methods ஆகவோ இருக்கும்.

14.15 Query-யை execute செய்தல்

Procedural முறையில் mysqli_query() function , ஒரு query-யை அதன் முதல் argument-ஆக எடுத்துக்கொண்டு ஒரு result handle-ஐக் கொடுக்கும். பின்னர் இந்தresult handle , இதனைத் தொடர்ந்து அடுத்தடுத்து வரும் “returned data”-வை உள்ளடக்கிய functions-க்கு argument-ஆக அனுப்பப்படும்.

```
$result = mysqli_query(query, $conn);

$result = $conn->query(query);
```

Object-oriented முறையில் query() function, ஒரு query-யை அதன் argument-ஆக எடுத்துக்கொண்டு result handle-ஐக் கொடுக்கும். இந்த result மட்டுமே ஒரு object-ஆக கருதப்படுவதால், இந்த object-ன் மீது அடுத்தடுத்து வரும் methods-ஐ அழைப்பதன் மூலம் நாம் queries-ஐ process செய்யலாம்.

14.16 Result set-லிருந்து தகவல்களைப் பெறுதல்

procedural முறையில் mysqli_fetch_array() எனும் function ஒவ்வொரு முறை அழைக்கப்படும்போதும், result-லிருந்து ஒரு data record-ஐக் கொடுக்கும்.
அதேபோல் object-oriented முறையில் result handle எனும் object-ன் மீது செயல்படும் fetch_array() function-ம் இதே வேலையைச் செய்கிறது.
இந்த இரண்டு முறையிலும் கடைசியில் data ஏதும் இல்லாத நிலையில் “NULL”-ஆனது return செய்யப்படும்.

```
$row = mysqli_fetch_array($result);

$result->fetch_array();
```

Array-ஆல் return செய்யப்படும் மதிப்புகள் numeric மற்றும் associative indexes-ஐப் பெற்றிருக்கும்.
Numeric index-ன் மதிப்புகள் 0-வில் தொடங்கி select செய்யப்பட்டுள்ள columns-ஐப் பொருத்து இடதிலிருந்து வலதுவரை அமையும். Associative indexes-ன் மதிப்புகள், column-ன் பெயர்கள் அல்லது query-ல் இருக்கும் aliases-ஐப் பொறுத்து அமையும்.

ஒரு query-ஆல் return செய்யப்படும் rows-ன் எண்ணிக்கையை அறிய procedural முறையில் mysqli_num_rows() எனும் function-ஐயும், object-oriented முறையில் result object-ன் மீது num_rows() எனும் attribute-ஐயும் பயன்படுத்தி கண்டறியலாம். அவ்வாறே mysqli_num_fields அல்லது num_fields என்பவை ஒரு query-ஆல் return செய்யப்படும் column-ன் எண்ணிக்கையை அறிய உதவும்.

14.17 Error Messages-ஐ வெளிப்படுத்துதல்

Procedural முறையில், mysqli_error() எனும் function, “Connection handle”-ஐ அதன் argument-ஆக எடுத்துக்கொண்டு, அந்த connection-ல் கடைசியாக execute செய்யப்பட்ட query-ல் இருக்கும் error பற்றிய தகவல்களைக் கொடுக்கும்.

```
mysqli_error($conn);
```

```
$conn->error();
```

Object-oriented முறையில் connection object-ன் மீது செயல்படும் error() method ஆனது இதே வேலையைச் செய்கிறது. இவ்வாறே error எண்ணைக் கண்டுபிடிக்க mysqli_errno() மற்றும் errno() என்பதைப் பயன்படுத்தலாம்.

14.18 இணைப்பை நிறுத்துதல்

ஒரு PHP script முடியும்போது MySQL resources எல்லாம் தானாகவே அழிக்கப்பட்டுவிடும்.

mysqli_close($conn);

$conn->close();

ஆனால், நீங்கள் இந்த resources-ஐ விடுவிக்க விரும்பினால் மேலே குறிப்பிட்டுள்ளவாறு procedural முறையில் mysqli_close() எனும் function-ம் object-oriented முறையில் conn object-ன் மீது செயல்படும் close() எனும் method-ம் செய்யும். Database இணைப்பை முடிக்காமல், அதன் result resource-ஐ மட்டும் விடுவிக்க விரும்பினால் procedural முறையில் mysqli_free_result() எனும் function மற்றும் object-oriented முறையில் result object-ன் மீது செயல்படும் free_result() எனும் method செய்யும்.

உதாரணம்

<?php

/* Establish a database connection */

$conn = new mysqli("localhost","user", "password", "dbname");

if (!$conn) {

echo "Failed to connect to database: Error " .

$conn->error(). "
\n";

exit;

}

/* Execute a query */

$sql_statement = "SELECT book_id, cond, title FROM book";

```
$result = $conn->query($sql_statement);
if (!$result) {
echo "Error executing query: Error: " .
$conn->error(). "<br>\n";
exit;
}
/* Loop through the result set to display it */
echo "<table>\n";
while ($row = $result->fetch_array()) {
echo "<tr>\n";
for ($i=0; $i<$result->num_rows; $i++) {
echo "<td>" . $row[$i] . "</td>\n";
}
echo "</tr>\n";
}
echo "</table>";
?>
```

15

Backup எடுத்தல் மற்றும் Trouble Shooting செய்தல்

நன்றாக இயங்கிக் கொண்டிருக்கும் MySQL Server-க்கு பல வகையான அபாயங்கள் ஏற்பட வாய்ப்பு உள்ளது. எனவே நாம் இந்தப் பகுதியில் எவ்வாறு அவ்வகையான அபாயங்களை எதிர்கொள்வது என்பது பற்றிப் பார்ப்போம்.

15.1 Backup எடுத்தல்

நமது database-ல் இருக்கும் தகவல்களை எல்லாம் எடுத்து வைத்துக் கொள்வது என்பது ஓர் அத்தியாவசியமான செயல் ஆகும். ஏனெனில் அபாயங்கள் ஏற்படும் காலங்களில், நமது data-வெல்லாம் அழிந்துவிட நேரிடின், இந்த backup எடுத்துவைக்கப்பட்ட file-லிலிருந்து நமது அழிந்த data-வை நாம் மீண்டும் பெற்றுக் கொள்ளலாம்.
ஒரு database-ல் இருக்கும் தகவல்களை backup எடுத்து வைப்பதற்கான கால இடைவெளி பல காரணிகளால் தீர்மானிக்கப்படுகிறது. அவை எவ்வளவு முறை ஒரு database-ஆனது update செய்யப்படுகிறது மற்றும் backup எடுக்கப்படும் database-ஐ offline-ல் வைப்பதன் மூலம் உண்டாகும் பிரச்சனைகளின் விளைவுகள் போன்றவற்றைப் பொருத்து அமையும்.
மேலும் backup எடுக்கப்படும் முறையினை இரண்டு வகைப்படுத்தலாம். அவை Full backup மற்றும் Incremental backup என்பதாகும். இவற்றைப் பற்றி விரிவாக பின்வருமாறு காணலாம்.

15.2 Full Backup எடுத்தல்

ஒரு database-ல் இருக்கும் தகவல்கள் அனைத்தையும் தொடர்ச்சியான sql file-ல் backup எடுத்து வைக்க mysqldump எனும் command பயன்படும்.

இதில் database-ல் இருக்கும் ஒவ்வொரு table-ம் ஒரு create statement-ஆகவும், அந்த table-ல் இருக்கும் ஒவ்வொரு row-வும் ஒரு insert statement ஆகவும் மாற்றப்பட்டு sql file-ல் சேமித்து வைக்கப்படுகிறது.

பொதுவாக இந்த mysqldump எனும் command அதன் output-ஐ திரையில் வெளிப்படுத்தும். எனவே இந்த output-ஐ ஒரு file-க்குள் redirect செய்வதன் மூலம் நாம் backup file-ஐ உருவாக்க முடியும்.
mysqldump –user = user –host = host –port = port \

–password = password dbname > filename.sql

mysqldump எனும் command-ஐத் தொடர்ந்து இணைப்புகளுக்குத் தேவையான விவரங்களைஎல்லாம் அளித்தபின், எந்த database-ஐ backup எடுக்க வேண்டுமோ அந்த database-ன் பெயரைக் குறிப்பிட வேண்டும்.

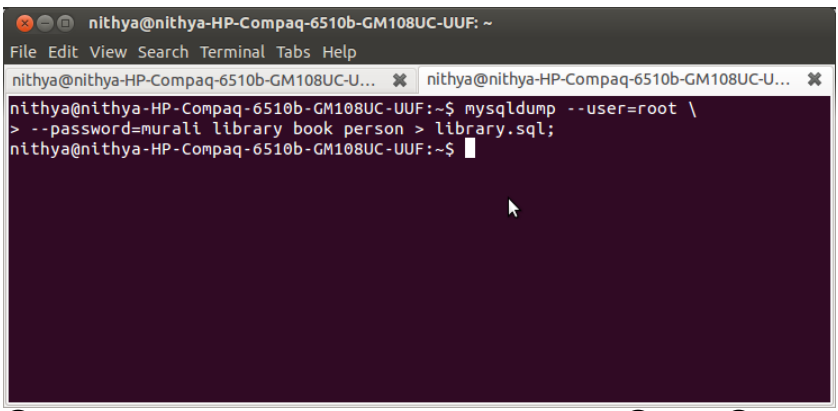
இந்த database-ஐத் தொடர்ந்து எந்த table-ன் பெயரும் கொடுக்கப்படவில்லையெனில், அந்த முழு table-ம் backup எடுத்துவைக்கப்படும். அப்படி இல்லாமல், அந்த database-ஐத் தொடர்ந்து ஏதேனும் ஒரு சில tables கொடுக்கப்பட்டிருப்பின், அந்த tables மட்டும் backup எடுத்து வைக்கப்படும்.

பின்வரும் எடுத்துக்காட்டில் library எனும் database-ஐத் தொடர்ந்து book மற்றும் person எனும் இரண்டு tables கொடுக்கப்பட்டிருப்பதால் இந்த இரண்டு tables மட்டும் library.sql எனும் file-ல் backup எடுத்துவைக்கப்படும். இந்த

command ‘shell’ prompt-ல் run செய்யப்பட வேண்டும்.

mysqldump –user = root \

–password = murali library book person > library.sql;

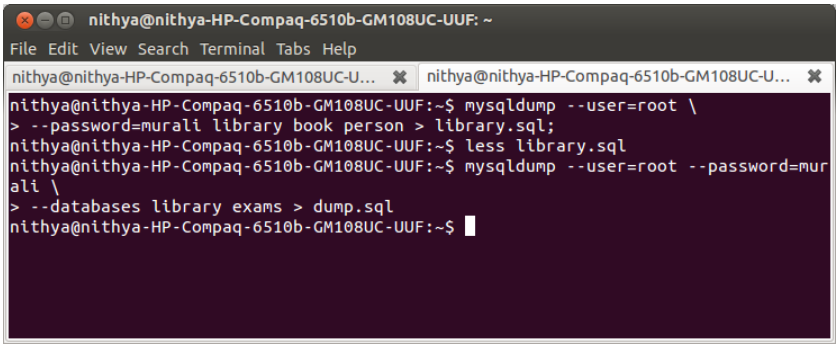


இந்த command-ல் host மற்றும் port எனும் இரண்டு options-ம் கொடுக்கப்படவில்லை. எனவே இந்த command, local machine-ல் default port-ஐப் பயன்படுத்தி இணைக்க முயற்சிக்கும்.

மேலும் நீங்கள் ஒன்றுக்கும் மேற்பட்ட databases-ஐ backup எடுக்க விரும்பினால் –databases எனும் switch-ஐப் பயன்படுத்தி, அதனைத் தொடர்ந்து நீங்கள் backup எடுக்க விரும்பும் databases-ன் பெயர்களைப் பட்டியலிடலாம். இது பின்வருமாறு.

mysqldump –user = root –password = murali \

–databases library exams > dump.sql



இது போன்றே, ஒரு MySQL server-ல் இருக்கும் அனைத்து databases-ஐயும் backup எடுக்க –all-databases எனும் switch ஐப் பயன்படுத்தலாம்.

15.3 Incremental backup எடுத்தல்

ஒரு database-ஐ முழுவதும் backup எடுத்து வைப்பது என்பது மிகுந்த நேரம் பிடிக்கும் ஒரு வேலை. ஏனெனில் full backup எடுக்கும்போது, ஒரு database-ல் இருக்கும் tables மற்றும் அந்த table-ல் இருக்கும் rows போன்ற அனைத்து விவரங்களும் mysqldump எனும் command-க்கு அனுப்பப்பட வேண்டும். இவ்வாறு செய்வதால் busy-யாக இயங்கிக் கொண்டிருக்கும் server-ன் performance பாதிக்கப்படுகிறது.

எனவே இதற்கு மாற்றாக இருப்பதுதான் Incremental backup என்பதாகும். இந்த Incremental backup-ல் MySQL server ஆனது ஒவ்வொருமுறை restart செய்யப்படும்போதும் அல்லது FLUSH LOGS எனும் command ஒவ்வொரு முறை வழங்கப்படும்போதும் binary logging feature மூலம் ஒரு புதிய binary log உருவாக்கப்படும்.

இந்த binary logging feature-ஐ enable செய்ய log_bin எனும் option-ஐ my.cnf file-ல் சேர்த்து, MySQL server-ஐ restart செய்யவும் அல்லது mysqld-ஐ –-log-bin எனும் switch-வுடன் சேர்த்து துவக்கவும்.

இவ்வாறு உருவாக்கப்பட்ட binary log-ஆனது ஒரு user-ன் home directory-ல் அதாவது பொதுவாக /var/lib/mysql எனும் பகுதியில், hostname-bin.XXX எனும் பெயரில் சேமித்து வைக்கப்படும். இதில் hostname என்பது ஒரு server-வுடைய முகவரியாகவும், XXX என்பது தொடர்ச்சியாக அமையும் எண்ணாகவும் இருக்கும்.

ஒவ்வொரு முறை MySQL server ஆனது restart செய்யப்படும்போதும் அல்லது ஒவ்வொருமுறை FLUSH LOGS எனும் command-ஐ வழங்கும்போதும் ஒரு புதிய binary log உருவாக்கப்படும் .

இந்த binary log-ஆனது update செய்யப்படுகின்ற data-க்கான SQL query-ஐ உள்ளடக்கியிருக்கும். எந்த ஒரு update, Insert அல்லது delete statement-ஆனாலும் இந்த binary log-ல் பதிவு செய்யப்படும்.

உதாரணத்துக்கு ஒரு update statement ஆனது எந்த ஒரு row-வையும் update செய்யவில்லை என்றாலும் அந்த statement இந்த log-ல் பதிவுசெய்யப்படும். இவ்வாறு பதிவு செய்யப்படுவதன் மூலம் நம்மால் ஒரு database-ன் full backup-லிருந்து சமீபத்தில் நிகழ்ந்த database மாற்றங்கள் வரை backup எடுத்து வைத்துக் கொள்ள முடியும்.

இந்த binary log-ஆனது full backup எடுக்கப்பட்டவுடன் நிகழும் transactions-வுடன் துவங்க வேண்டும். இந்த binary log-ஐ backup file-வுடன் synchronize செய்ய, mysqldump எனும் command-வுடன் –flush-logs எனும் option-ஐ பயன்படுத்தவும். இது dump துவங்கும்போதே FLUSH LOGS command-ஐ வழங்கப் பயன்படும். பின்னர் database-ல் நடைபெறும் செயல்கள் அனைத்தும் அடுத்தடுத்த binary log-ல் தொடர்ச்சியாக எழுதப்படும்.

இவ்வாறு binary logs-ஐ எழுதுவதன் மூலம் server-வுடைய performance பாதிக்கப்பட்டாலும் அந்த performance இழப்பு மிகச் சிறிய அளவில் அமையும். MySQL documentation-ல் binary logging, enable செய்யப்பட்ட ஒரு server-ன் performance குறைவாக இருக்கும் எனக் கூறப்பட்டுள்ளது

15.4 Backup-ஐ restore செய்தல்

திடீரென்று நமது database-ல் நாம் சேமித்து வைத்த தகவல்களெல்லாம் அழிந்து விட்டதெனில், ஏற்கனவே நாம் backup எடுத்து வைத்திருக்கும் file-லிலிருந்து அதே தகவல்களை மீண்டும் எவ்வாறு restore செய்வது என்று பார்க்கலாம்.

15.5 Full backup-ஐ restore செய்தல்

mysql –user = user –host = host –port = port \

–password = password dbname < filename.sql

மேலே குறிப்பிட்டுள்ளவாறு backup எடுத்துவைக்கப்பட்ட file-ஐ MySQL எனும் command-க்கு input file-ஆக அமைப்பதன் மூலம் சுலபமாக நாம் ஒரு database-ஐ restore செய்யலாம்

நீங்கள் backup எடுக்கப்பட்ட database-ன் பெயரையே மீண்டும் பயன்படுத்தி அதே பெயரிலேயே மீண்டும் backup file-ஐ restore செய்தீர்களானால், அந்த database-ல் ஏற்கனவே இருக்கும் tables அனைத்தும் overwrite செய்யப்படும்.

mysqldump-ன் output-ஆனது, backup எடுக்கப்படும்

ஒவ்வொரு table-க்கும் ஒரு Drop Table statement-ஐ உள்ளடக்கியிருக்கும்.

ஒரே ஒரு database-க்கான backup file-ஆனது எந்த ஒரு Create Database statement-ஐயும் உள்ளடக்கியிருக்காது. எனவே நீங்கள் இவ்வகையான file-ஐ restore செய்யும்போது MySQL command-ல் அதே database-ன் பெயரை மீண்டும் பயன்படுத்தாமல் வேறொரு database-ன் பெயரைப் பயன்படுத்தி restore செய்ய வேண்டும்.

பல database-க்கான backup file-ஆனது mysqldump எனும் command-வுடன் –all-databases அல்லது –databases எனும் option-ஐப் பயன்படுத்தி உருவாக்கப்படும். இந்த file-ல் இருக்கும் ஒவ்வொரு database-க்கும் Create Database எனும் command ஆனது தானாகவே சேர்க்கப்பட்டிருக்கும்

15.6 Incremental Backup-ஐ restore செய்தல்

நீங்கள் ஒரு database-க்கான Incremental backup-ஐ restore செய்வதற்கு முன்னர், அதன் full backup-ஐ restore செய்ய வேண்டும். அதற்கு அடுத்தபடியாகத்தான் அந்த database-ல் நிகழ்த்த மாற்றங்கள் அனைத்தையும் உள்ளடக்கிய binary log files-ஐ முறைப்படி restore செய்ய வேண்டும்.

இதற்காக mysqlbinlog எனும் utility பயன்படுத்தப்படும். இந்த utility ஆனது binary log format-ல் இருக்கும் ஒரு backup file-ஐ sql commands-ஐ உள்ளடக்கிய text format-ல் மாற்றும். பின்னர் இந்த mysqlbinlog-ன் output ஆனது MySQL command-க்கு input-ஆக அனுப்பப்படும்.

mysqlbinlog hostname-bin.XXX | mysql –options dbname
இவ்வாறு நாம் ஒரு command-ன் output-ஐ மற்றொரு command-க்கு input-ஆக அனுப்ப pipe-மூலமாகவோ அல்லது file re-direction மூலமாகவோ செய்யலாம்.

பின்வரும் command, ஒரு database-ல் இருக்கும் ஒவ்வொரு binary log file-ஐயும் regular expression-ஐப் பயன்படுத்துவதன் மூலம் ஒரே command-ல் restore செய்வதைக் காணலாம்.

mysqlbinlog hostname-bin.[0-9]* | \

mysql –options dbname

இந்த regular expression, கொடுக்கப்பட்டுள்ள format-ல் பொருந்தும் ஒவ்வொரு file-ஐயும், இந்த command கண்டுபிடித்து அதன் ஒவ்வொன்றாக MySQL-க்கு input-ஆக அனுப்பும். ஏனெனில் இந்த file name-ல் இருக்கும் எண்கள் எல்லாம் முன்னால் 0-ஆல் நிரப்பப்பட்டிருப்பதால், இந்த command அந்த file-ஐ தொடர்ச்சியான முறையில் read செய்யும். நீங்கள் ஒன்றுக்கும் மேற்பட்ட binary log files-ஐ manual-ஆன முறையில் restore செய்ய விரும்பினால், முறையான எண் வரிசையில் அந்த file-ஐப் பயன்படுத்த வேண்டும்.

15.7 Table-ல் இருக்கும் corrupt-ஆன data-வைக் கண்டுபிடித்தல்

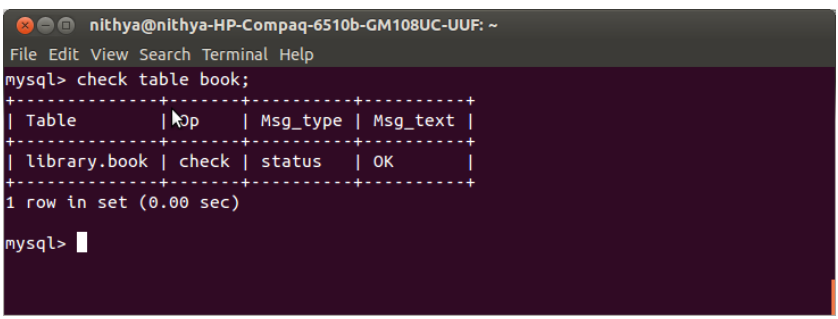
அரிதாக MySQL data files-ல் இருக்கும் தகவல்கள் corrupt ஆக்கப்பட்டு repair செய்யப்படும் நிலைமையில் இருக்கும் . எனவே ஒரு table எவ்வகையான நிலைமையில் உள்ளது என்பதை CHECK TABLE எனும் command-ஐப் பயன்படுத்தி தெரிந்து கொள்ளலாம். இதற்கான syntax பின்வருமாறு.

CHECK TABLE table;

REPAIR TABLE table;
பின்வரும் command, “book” எனும் table-ல் ஏதேனும் error உள்ளதா என்பதைக் கண்டுபிடிக்கப் பயன்படும்.

CHECK TABLE book;

இந்த command-ன் output பின்வருமாறு அமைந்தால், இந்த table நல்ல நிலையில் உள்ளது என்று அர்த்தம். எனவே நாம் பழுதுபார்க்கும் வேலையாவும் இந்த table-ல் செய்யத் தேவையில்லை.



சிலசமயங்களில் இந்த output-ல் இருக்கும் msg_text எனும் column-ஆனது check செய்யப்படும் table-ல் ஏதேனும் error இருந்தால் அந்த error-ஐ வெளிப்படுத்தும். இவ்வாறு error ஏதும் வெளிப்படுத்தப் பட்டிருப்பின் REPAIR TABLE எனும் command மூலம் MySQL ஆனது அந்த error-ஐ சரிசெய்ய முயற்சிக்கும்.

மேலும் MySQL-வுடன் வரும் myisamchk எனும் utility, MyISAM tables-ஐ SQL commands-ஐ விடவும் சிறந்த முறையில் சோதித்து பழுது பார்க்கும். வெறும் myisamchk எனும் command எந்த option-ம் இல்லாமல் run செய்யப்படும்போது, அது ஒரு table-ல் ஏதேனும் error உள்ளதா என்று மட்டும் சோதிக்கும். மேலும் பல வகையான Command line switches-ஐ இணைப்பதன் மூலம் இன்னும் சில தகவல்களை நாம் பெற முடியும்.

இந்த myisamchk எனும் utility-ஆல் சோதிக்கப்படும்

MyISAM table files அனைத்தும் நமது data directory-ல் .MYI file எனும் extension-வுடன் சேமித்து வைக்கப்பட்டிருக்கும். நமது data directory-ஆனது /var/lib/mysql, எனில் “sampdb” எனும் database-ல் இருக்கும் tables அனைத்தும் பின்வரும் command மூலம் சோதிக்கப்படும்.

myisamchk /var/lib/mysql/sampdb/*.MYI

பொதுவாக இந்த command-வுடன் –fast எனும் switch-ஐ இணைப்பதன் மூலம் அனைத்து tables-ஐயும் நாம் விரைவாக சோதித்து விடலாம். இது எந்தெந்த tables முறையாக மூடப்படவில்லை என்பதை மட்டும் சோதிக்கும்.

மேலும் –medium-check எனும் option முறையாக அனைத்து விதமான சோதனைகளையும் table-ல் சோதித்து விட்டு பெரும்பாலான errors-ஐக் கண்டுபிடிக்கும். இந்த –medium-check எனும் option-ஆல் பிரச்சனையைக் கண்டுபிடித்து தெளிவாகக் கூற முடியவில்லை என்றால் மட்டுமே –extend-check எனும் command ஐப் பயன்படுத்தவும். ஏனெனில் இந்த option மிகவும் மெதுவாக இயங்கும், ஆனால் இன்னும் தெளிவான முறையில் table-ஐ ஆராய்ந்து பிரச்சனையைக் கண்டுபிடித்து கூறிவிடும்.

இவ்வாறாக மேற்கூறிய அனைத்து switches-ஐயும் பயன்படுத்தி, ஒரு table file எவ்வாறு corrupt ஆகியுள்ளது என்பதைக் கண்டறிந்தவுடன், –recover எனும் switch மூலம் நாம் அதனைப் பழுது பார்க்கலாம். ஆனால் பழுதுபார்ப்பதற்கு முன்னர் நாம் mysqld-ஐ நிறுத்த வேண்டும். ஏனெனில் நாம் பழுதுபார்க்கும் சமயத்தில் எந்த ஒரு data-வும் MySQL server-ஆல் நாம் repair செய்து கொண்டிருக்கும் file-ல் எழுதப்படக்கூடாது.

சிலசமயங்களில் –recover எனும் switch மூலம் நமது data file-ல் இருக்கும் error-ஐ சரிசெய்ய

முடியவில்லையெனில், –safe-recover எனும் switch மூலம் நாம் சரிசெய்யலாம். இது மிகவும் மெதுவாக செயல்பட்டாலும் error-ஐ கண்டுபிடித்து சரிசெய்துவிடும்

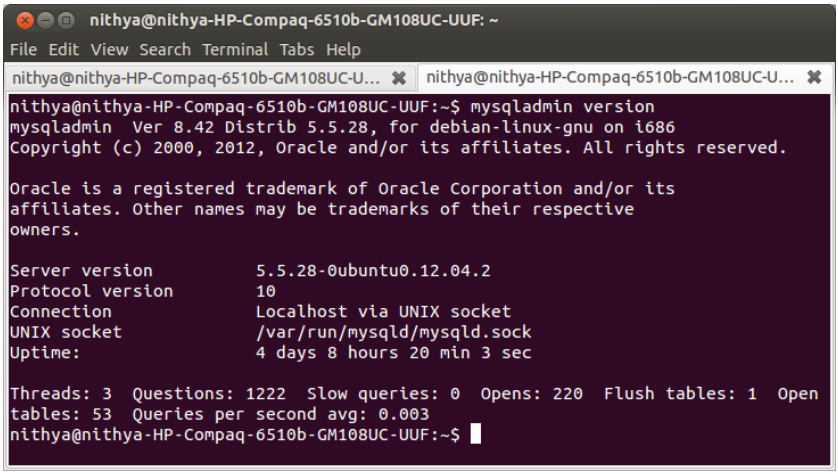
15.8 Server Crash-ஐ கண்டுபிடித்தல்

என்னதான் MySQL முழுவதுமாக பரிசோதிக்கப் பட்டாலும் சிலசமயங்களில் அதன் server-ல் crash ஏற்பட வாய்ப்பு உள்ளது. அத்தகைய சமயங்களில் நாம் இந்த server crash-ஐப் பற்றி ஒரு bug report -ஐ உருவாக்க வேண்டும். இதைப்பற்றி பின்வருமாறு காணலாம்.

முதலில் ஒரு crash ஏற்படும்போது அதற்கு MySQL server காரணமா அல்லது அதன் client காரணமா என்று கண்டுபிடிக்க வேண்டும். இதற்காக,

mysqladmin version

எனும் command-ஐ run செய்யவும். இது எவ்வளவு நேரமாக UP-ல் உள்ளது எனும் விவரத்தைக் கொடுக்கும். இதன் OUTPUT பின்வருமாறு.



Uptime: 4 days 8 hours 20 min 3 sec

இந்த output-ன் மூலம் server-ஆனது நீண்ட நேரமாக UP நிலையில் உள்ளது என்பதை அறியலாம். எனவே பிரச்சனைக்குக் காரணம் server அல்ல client-தான் எனத் தீர்மானித்து நமது client program-ஐ சரிசெய்தால் போதுமானது.

ஒருவேளை இந்த output-ஆல் தெரிவிக்கப்படும் UP time மிகக் குறைவாக இருந்தாலோ அல்லது mysqld ஆனது crash-க்குப் பின்னர் இயங்காமல் இருந்தாலோ, முதலில் நாம் நமது tables அனைத்தையும் corrupt-ஆகாமல் உள்ளதா என்று சோதிக்க வேண்டும். இதற்காக நாம் mysqld -ஐ நிறுத்திவிட்டு பின்வரும் command-ஐ data directory-ல் இருந்து run செய்யவும்.

myisamchk –silent –force */*.MYI

இது நமது system-ல் இருக்கும் ஒவ்வொரு table-ஐயும் சோதிக்கும். பின்னர் mysqld-ஐ –log எனும் switch-வுடன் சேர்த்து restart செய்யவும், அல்லது my.cnf file-ல் [mysqld] பகுதியின் கீழ் log-ஐ இணைக்கவும். இது execute செய்யப்படும் ஒவ்வொரு sql command-ஐயும் file-ல் log

செய்யும். மேலும் நாம் அதே commands-ஐ மீண்டும் மீண்டும் பயன்படுத்தி crash-ஐ தொடர்ச்சியாக உருவாக்க முயற்சிக்க வேண்டும்.

பின்னர் நீங்கள் http://bugs.mysql.com/ எனப்படும் bug database-ஐ query செய்து, அதன் பின்னர் bug report-ஐ file செய்யலாம்.

மேலும் crash-க்கான மற்றொரு காரணம் server-இடம் கூட இருக்கலாம். ஏனெனில் நம்முடைய system-ன் RAM-ல் பிரச்சனை என்றாலோ அல்லது hard drives-ல் பிரச்சனை என்றாலோ அது MySQL-ல் crash-ஐ உண்டாக்கும். எனவே நமது server crash-க்கு சரியான காரணம் கிடைக்கவில்லையெனில் நம்முடைய system-ன் hardware-ஐயும் ஒருமுறை முழுசாகப் பரிசோதிப்பது நல்லது.

15.9 ஒரு சில பொதுவான ERRORS

இந்தப் பகுதியில் நாம் அடிக்கடிப் பார்க்கும் ஒருசில பொதுவான errors-ஐயும் மற்றும் அதனை எவ்வாறு கையாளுவது என்பது பற்றியும் காணலாம்.

Can’t Connect to MySQL Server

பொதுவாக இந்த error, server machine-ல் mysqld process இயங்கவில்லை என்பதைக் குறிக்கும்.

ERROR 2002: Can’t connect to local MySQL server

through socket ‘/var/lib/mysql/mysql.sock’ (2)

ERROR 2003 (HY000): Can’t connect to MySQL server

on ’123.45.67.89′ (113)

நீங்கள் உங்களுடைய local MySQL server-வுடன் இணைக்க முடியவில்லையெனில், உங்கள் system-ல் இருக்கும் process list-ஐப் பார்க்கவும். Linux-ல் ps -ax எனும் command-ம், Windows-ல் Task Manager எனும் command-ம், உங்கள் system-ல் இயங்கும் process-ஐப் பட்டியலிடப் பயன்படும். இதில் mysqld process பட்டியலிடப்படவில்லை எனில்,உங்கள் MySQL server-ஐ start செய்யவும்.

அடுத்தபடியாக, socket file நீக்கப்படும்போதும் இவ்வகையான பிரச்சனை ஏற்பட வாய்ப்பு உள்ளது. பொதுவாக இந்த socket file, /tmp/mysql.sock எனும் பகுதியில் இருக்கும். சில systems பொதுவாக /tmp எனும் பகுதியில் இருப்பவற்றைஎல்லாம் ஒரு குறிப்பிட்ட கால இடைவெளியில் அழித்து சுத்தப்படுத்திக் கொண்டே இருக்கும். மேலும் இந்த /tmp எனும் பகுதியை எந்த user வேண்டுமானாலும் அணுகலாம். எனவே இங்கு இருக்கும் mysql.sock எனும் முக்கியமான file-ஐ யார் வேண்டுமானாலும் அழிக்க வாய்ப்பு உள்ளது. எனவே இந்த socket file-ஐ MySQL user-க்கு மட்டுமே அனுமதியிருக்கும் மற்றொரு directory-க்கு மாற்றிவிடுவது இந்தப் பிரச்சனையைத் தவிர்க்கும். இதற்காக பின்வரும் வரிகளை my.cnf file-ல் இணைக்கவும்.

[mysqld]

socket = /path/to/mysql.sock

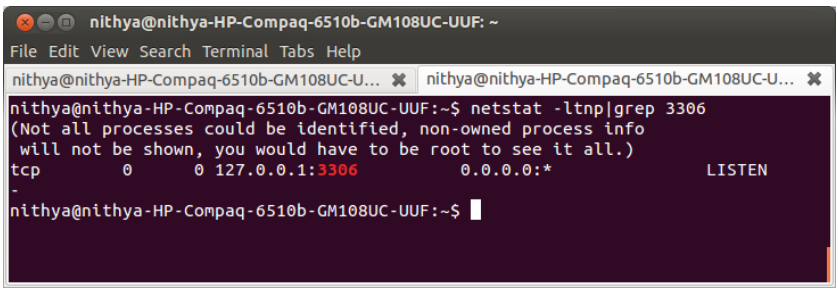
[client]

socket = /path/to/mysql.sock

நாம் எதிர்பார்க்கும் Port-ல் தான் mysqld ஆனது network இணைப்புகளைப் பெற்றுக்கொள்கிறதா என்பதை சோதிக்க Unix/Linux-ல் பின்வரும் command

பயன்படுகிறது.

shell > netstat -ltnp | grep 3306



மற்றும் windows-ல் இது பின்வருமாறு பயன்படுத்தப்படுகிறது.

> netstat a | find “LISTENING”

உங்களால் remote MySQL server-வுடன் இணையமுடியவில்லை, ஆனால் mysqld-ஆனது இயங்கிக்கொண்டுதான் இருக்கிறது எனத் தெரிந்தால் firewall settings-ஐ சரிபார்க்கவும். Port 3306 அல்லது எந்த port இணைப்பிற்காகப் பயன்படுத்தப்படுகிறதோ அந்த port-ல் TCP/IP traffic-ஐ அனுமதிக்க வேண்டும்.

இந்தப் பகுதியில் நாம் பார்த்த error message ஆனது “MySQL server-வுடன் இணைய முடியவில்லை” என்பதைக் குறிக்கிறதே தவிர “இணைய அனுமதி இல்லை” என்று குறிப்பிடவில்லை. எப்பொழுது நீங்கள் MySQL server-வுடன் இணையும்போது தவறான password-ஐ அளிக்கிறீர்களோ அப்போதுதான் “உங்களுக்கு அனுமதி இல்லை” எனும் error வரும். இதைப்பற்றி பின்வருமாறு காண்போம்

Access Denied

நாம் அளிக்கும் username அல்லது password தவறாக இருந்தாலோ அல்லது நாம் பயன்படுத்த விரும்பும் database-க்கு நாம் அளிக்கும் username அனுமதிகளைப் பெற்றிருக்கவில்லை என்றாலோ இவ்வகையான error வரும்.

ERROR 1045 (28000): Access denied for user

‘user’@’host’ (using password: YES)

அப்போது நீங்கள் MySQL database-ல் இருக்கும் privilege table-ஐ பயன்படுத்தி, நாம் இணைப்பிற்காகக் கொடுத்துள்ள connection arguments-யாவும் சரியாக உள்ளதா என சரிபார்த்துக் கொள்ளவும்

நாம் –password என்பதற்கு பதிலாக -p எனும் switch-ஐப் பயன்படுத்தினால் நாம் அளிக்கும் password-ஐத் தொடர்ந்து எந்த ஒரு இடைவெளியும் கொடுக்கக்கூடாது. அவ்வாறு கொடுத்தால் -p-ஐத் தொடர்ந்துவரும் argument ஆனது database name ஆகக் கருதப்படும்.

Too Many Connections

MySQL-ல் நாம் எவ்வளவு அதிகமான தொடர் இணைப்புகளை உருவாக்க முடியும் எனும் விவரம் max_connections எனும் variable-ல் சேமித்து வைக்கப்பட்டிருக்கும். பொதுவாக இதன் மதிப்பு 100 என்று இருக்கும். எனவே எப்போது நாம் 100-க்கும் மேற்பட்ட இணைப்புகளை உருவாக்க முயற்சிக்கிறோமோ அப்போது இந்த error வரும்.

ERROR 1040: Too many connections

எனவே நாம் விரும்பும் புதிய மதிப்பினை இந்த variable-க்கு my.cnf file-ல் பின்வருமாறு அளிப்பதன் மூலம் இந்த errors-ஐ தவிர்க்கலாம்.

[mysqld]

max_connections = 200

ஆனால் இவ்வாறு செய்வது ஒரு நல்ல தீர்வு ஆகாது. எப்பொழுது MySQL, அதிகபட்சமான இணைப்புகளின் எண்ணிக்கையைத் தாண்டிச் செல்கிறதோ அப்போது அதற்கான காரணத்தை நாம் கண்டுபிடிக்க வேண்டும். நமது system அதிகப்படியான இணைப்புகளைத் தாங்கும் என்று தெரிந்தால் மட்டுமே இந்த max_connections-ன் மதிப்பினை நாம் அதிகரிக்கலாம்.

உண்மையில் இந்த variable-ஆனது நாம் கொடுக்கும் அதிகபட்ச எண்ணிக்கையைத் தாண்டி இன்னும் ஒரு இணைப்பினை அனுமதிக்கும். இந்த இணைப்பு superuser-ஆல் பயன்படுத்தப்படும். எப்பொழுது இணைப்புகள் அதிகபட்ச எண்ணிக்கையைத் தாண்டுகிறதோ அப்பொழுது இந்த இணைப்பு super user-க்குப் பயன்படும் வகையில் அமையும்.

மேலும் mysqladmin processlist எனும் command-ஐப் பயன்படுத்தி ஏன் இவ்வாறு இணைப்புகள் அதிக எண்ணிக்கையில் உள்ளன என்று கண்டுபிடிக்கலாம்.

MySQL Server Has Gone Away

mysqld-ஆனது அதன் client-வுடன் கொண்டுள்ள இணைப்பினை துண்டித்து அதன் நேரம் முடியும்போது இந்த error வரும்.

ERROR 2006: MySQL server has gone away

பொதுவாக இந்த இணைப்பு துண்டிக்கப்படுவதற்கான கால இடைவெளி 8 மணி நேரம். வேண்டுமானால் நாம் wait_timeout எனும் system variable-ஐப் பயன்படுத்தி இந்த மதிப்பினை மாற்றலாம். இந்த variable-ல் நேரம் seconds-ல் சேமிக்கப்பட்டிருக்கும். எப்பொழுது இந்த error, நேரம் முடிவதினால் வருகிறதோ அப்பொழுது mysql program தானாகவே மீண்டும் இணைப்பை உருவாக்க முயற்சிக்கும்.

இப்பொழுது அதே error திரும்ப வந்தால், mysqld process நிறுத்தப்பட்டுள்ளது என்று அர்த்தம். எனவே mysql-ஆல் மீண்டும் இணைக்கமுடியவில்லைஎனில், mysqld இயங்கிக் கொண்டிருக்கிறதா என்பதை சரிபார்க்கவும்.

Got Error from Table Handler

இந்த error நமது database-ன் table storage file-ல் பிரச்சனை உள்ளதைக் குறிக்கும்.

Error 1030: Got error 141 from table handler

பொதுவாக இவ்வகையான பிரச்சனை எல்லாம் myisamchk எனும் utility-ஆல் சரிசெய்யப்படும். இந்த message-ல் உள்ள error எண் , பிரச்சனையின் தன்மையைக் குறிக்க உதவுகிறது. இந்த error

எண்ணைப் பயன்படுத்தி perror command மூலம் அந்த எண்ணுக்கு தொடர்பான error message-ஐ பின்வருமாறு கண்டறியலாம்.

shell> perror 141

141 = Duplicate unique key or constraint on write

or update

எனவே இவ்வகையான error வரும்போது myaqld-ஐ நிறுத்திவிட்டு myisamchk — recover என்பதை அதன் தொடர்புடைய .MYI file-ல் run செய்யவும்.

சிலசமயம் இதனால் தெரிவிக்கப்படும் error message, system பிரச்சனைகளைச் சார்ந்ததாகக் கூட இருக்கலாம். இது பின்வருமாறு.

$ perror 28

Error code 28: No space left on device

15.10 உதவியை நாடுதல்

சில சமயம் நாம் பிரச்சனைகளைத் தீர்ப்பதற்கு சிலவற்றின் உதவியை நாட வேண்டியிருக்கும். முதலில் http://dev.mysql.com/doc/refman/5.0/en/temp0105.html எனும் பகுதியில் இருக்கும் online manual-ல் உங்களுக்குத் தேவையான பகுதியைப் படித்துத் தெரிந்து கொள்ளவும்.

பலவாறான MySQL mailing list-ஆனது பல்வேறு topics-ஐப் பற்றிய விவரங்களைக் கொடுக்கும். மேலும் இந்த lists-ஆனது பல்வேறு இடத்தில இருக்கும் பயனர் குழுக்களாலும் பல்வேறு மொழிகளாலும் பயன்படுத்தக்கூடிய வகையில் அமையும். சமீபத்திய mailing list-ஐப் பற்றி அறிய http://lists.mysql.com/ எனும்

முகவரியை நாடவும்.

மேலும் பிரச்சனைகள் ஏற்படும்போது நாம் அதற்குத் தேவையான உதவியை MySQL AB-இடம் இருந்து கூட பெறலாம். இந்த MySQL AB என்பது, வருடத்துக்கு ஒரு குறிப்பிட்ட தொகையில் support-ஐ அளிக்கும். அனைத்து support packages-ம் MySQL knowledge base-ஐ அணுகுவதற்கான அனுமதியை உள்ளடக்கியிருக்கும். இந்த MySQL Knowledge base என்பது நமக்குத் தேவையான விடைகளை உடனடியாகக் கொடுக்க உதவும் technical articles-ஐ உள்ளடக்கிய ஒரு searchable library ஆகும்

இதற்கு அடுத்தபடியாக 24 மணி நேரமும் , 7 நாட்களும் support அளிக்கக்கூடிய வகையில் தொலைபேசி சேவையும் உள்ளது. இதில் அவசர காலங்களில் உடனடியாக விடையளித்தல் மற்றும் MySQL expert மூலம், remote-ல் பிரச்சனையைத் தீர்த்தல் போன்ற பல்வேறு வகையான சேவைகளும் அடங்கும். மேலும் விவரங்களுக்கு www.mysql.com/support/ எனும் முகவரியின் உதவியை நாடவும்.

Free Tamil Ebooks - எங்களைப் பற்றி

மின்புத்தகங்களைப் படிக்க உதவும் கருவிகள்:

மின்புத்தகங்களைப் படிப்பதற்கென்றே கையிலேயே வைத்துக் கொள்ளக்கூடிய பல கருவிகள் தற்போது சந்தையில் வந்துவிட்டன. Kindle, Nook, Android Tablets போன்றவை இவற்றில் பெரும்பங்கு வகிக்கின்றன. இத்தகைய கருவிகளின் மதிப்பு தற்போது 4000 முதல் 6000 ரூபாய் வரை குறைந்துள்ளன. எனவே பெரும்பான்மையான மக்கள் தற்போது இதனை வாங்கி வருகின்றனர்.

ஆங்கிலத்திலுள்ள மின்புத்தகங்கள்:

ஆங்கிலத்தில் லட்சக்கணக்கான மின்புத்தகங்கள் தற்போது கிடைக்கப் பெறுகின்றன. அவை PDF, EPUB, MOBI, AZW3. போன்ற வடிவங்களில் இருப்பதால், அவற்றை மேற்கூறிய கருவிகளைக் கொண்டு நாம் படித்துவிடலாம்.

தமிழிலுள்ள மின்புத்தகங்கள்:

தமிழில் சமீபத்திய புத்தகங்களெல்லாம் நமக்கு மின்புத்தகங்களாக கிடைக்கப்பெறுவதில்லை. ProjectMadurai.com எனும் குழு தமிழில் மின்புத்தகங்களை வெளியிடுவதற்கான ஓர் உன்னத சேவையில் ஈடுபட்டுள்ளது. இந்தக் குழு இதுவரை வழங்கியுள்ள தமிழ் மின்புத்தகங்கள் அனைத்தும் PublicDomain-ல் உள்ளன. ஆனால் இவை மிகவும் பழைய புத்தகங்கள்.

சமீபத்திய புத்தகங்கள் ஏதும் இங்கு கிடைக்கப்பெறுவதில்லை.

எனவே ஒரு தமிழ் வாசகர் மேற்கூறிய “மின்புத்தகங்களைப் படிக்க உதவும் கருவிகளை” வாங்கும்போது, அவரால் எந்த ஒரு தமிழ் புத்தகத்தையும் இலவசமாகப் பெற முடியாது.

சமீபத்திய புத்தகங்களை தமிழில் பெறுவது எப்படி?

சமீபகாலமாக பல்வேறு எழுத்தாளர்களும், பதிவர்களும், சமீபத்திய நிகழ்வுகளைப் பற்றிய விவரங்களைத் தமிழில் எழுதத் தொடங்கியுள்ளனர். அவை இலக்கியம், விளையாட்டு, கலாச்சாரம், உணவு, சினிமா, அரசியல், புகைப்படக்கலை, வணிகம் மற்றும் தகவல் தொழில்நுட்பம் போன்ற பல்வேறு தலைப்புகளின் கீழ் அமைகின்றன.

நாம் அவற்றையெல்லாம் ஒன்றாகச் சேர்த்து தமிழ் மின்புத்தகங்களை உருவாக்க உள்ளோம்.

அவ்வாறு உருவாக்கப்பட்ட மின்புத்தகங்கள் Creative Commons எனும் உரிமத்தின் கீழ் வெளியிடப்படும். இவ்வாறு வெளியிடுவதன் மூலம் அந்தப் புத்தகத்தை எழுதிய மூல ஆசிரியருக்கான உரிமைகள் சட்டரீதியாகப் பாதுகாக்கப்படுகின்றன. அதே நேரத்தில் அந்த மின்புத்தகங்களை யார் வேண்டுமானாலும், யாருக்கு வேண்டுமானாலும், இலவசமாக வழங்கலாம்.

எனவே தமிழ் படிக்கும் வாசகர்கள் ஆயிரக்கணக்கில் சமீபத்திய தமிழ் மின்புத்தகங்களை இலவசமாகவே பெற்றுக் கொள்ள முடியும்.

தமிழிலிருக்கும் எந்த வலைப்பதிவிலிருந்து வேண்டுமானாலும் பதிவுகளை எடுக்கலாமா?

கூடாது.

ஒவ்வொரு வலைப்பதிவும் அதற்கென்றே ஒருசில அனுமதிகளைப் பெற்றிருக்கும். ஒரு வலைப்பதிவின் ஆசிரியர் அவரது பதிப்புகளை “யார் வேண்டுமானாலும் பயன்படுத்தலாம்” என்று குறிப்பிட்டிருந்தால் மட்டுமே அதனை நாம் பயன்படுத்த முடியும்.

அதாவது “Creative Commons” எனும் உரிமத்தின் கீழ் வரும் பதிப்புகளை மட்டுமே நாம் பயன்படுத்த முடியும்.

அப்படி இல்லாமல் “All Rights Reserved” எனும் உரிமத்தின் கீழ் இருக்கும் பதிப்புகளை நம்மால் பயன்படுத்த முடியாது.

வேண்டுமானால் “All Rights Reserved” என்று விளங்கும் வலைப்பதிவுகளைக் கொண்டிருக்கும் ஆசிரியருக்கு அவரது பதிப்புகளை “Creative Commons” உரிமத்தின் கீழ் வெளியிடக்கோரி நாம் நமது வேண்டுகோளைத் தெரிவிக்கலாம். மேலும் அவரது படைப்புகள் அனைத்தும் அவருடைய பெயரின் கீழே தான் வெளியிடப்படும் எனும் உறுதியையும் நாம் அளிக்க வேண்டும்.

பொதுவாக புதுப்புது பதிவுகளை உருவாக்குவோருக்கு அவர்களது பதிவுகள் நிறைய வாசகர்களைச் சென்றடைய வேண்டும் என்ற எண்ணம் இருக்கும். நாம் அவர்களது படைப்புகளை எடுத்து இலவச மின்புத்தகங்களாக வழங்குவதற்கு நமக்கு அவர்கள் அனுமதியளித்தால், உண்மையாகவே அவர்களது படைப்புகள் பெரும்பான்மையான மக்களைச் சென்றடையும். வாசகர்களுக்கும் நிறைய புத்தகங்கள் படிப்பதற்குக் கிடைக்கும்

வாசகர்கள் ஆசிரியர்களின் வலைப்பதிவு முகவரிகளில் கூட அவர்களுடைய படைப்புகளை தேடிக் கண்டுபிடித்து படிக்கலாம். ஆனால் நாங்கள் வாசகர்களின் சிரமத்தைக் குறைக்கும் வண்ணம் ஆசிரியர்களின் சிதறிய வலைப்பதிவுகளை ஒன்றாக இணைத்து ஒரு முழு மின்புத்தகங்களாக உருவாக்கும் வேலையைச் செய்கிறோம். மேலும் அவ்வாறு உருவாக்கப்பட்ட புத்தகங்களை “மின்புத்தகங்களைப் படிக்க உதவும் கருவிகள்”-க்கு ஏற்ற வண்ணம் வடிவமைக்கும் வேலையையும் செய்கிறோம்.

FreeTamilEbooks.com

இந்த வலைத்தளத்தில்தான் பின்வரும் வடிவமைப்பில் மின்புத்தகங்கள் காணப்படும்.

PDF for desktop, PDF for 6” devices, EPUB, AZW3, ODT

இந்த வலைதளத்திலிருந்து யார் வேண்டுமானாலும் மின்புத்தகங்களை இலவசமாகப் பதிவிறக்கம்(download) செய்து கொள்ளலாம்.

அவ்வாறு பதிவிறக்கம்(download) செய்யப்பட்ட புத்தகங்களை யாருக்கு வேண்டுமானாலும் இலவசமாக வழங்கலாம்.

இதில் நீங்கள் பங்களிக்க விரும்புகிறீர்களா?

நீங்கள் செய்யவேண்டியதெல்லாம் தமிழில் எழுதப்பட்டிருக்கும் வலைப்பதிவுகளிலிருந்து பதிவுகளை
எடுத்து, அவற்றை LibreOffice/MS Office போன்ற wordprocessor-ல் போட்டு ஓர் எளிய மின்புத்தகமாக மாற்றி எங்களுக்கு அனுப்பவும்.

அவ்வளவுதான்!

மேலும் சில பங்களிப்புகள் பின்வருமாறு:

1. ஒருசில பதிவர்கள்/எழுத்தாளர்களுக்கு அவர்களது படைப்புகளை “Creative Commons” உரிமத்தின்கீழ் வெளியிடக்கோரி மின்னஞ்சல் அனுப்புதல்
2. தன்னார்வலர்களால் அனுப்பப்பட்ட மின்புத்தகங்களின் உரிமைகளையும் தரத்தையும் பரிசோதித்தல்
3. சோதனைகள் முடிந்து அனுமதி வழங்கப்பட்ட தரமான மின்புத்தகங்களை நமது வலைதளத்தில் பதிவேற்றம் செய்தல்

விருப்பமுள்ளவர்கள் freetamilebooksteam@gmail.com எனும் முகவரிக்கு மின்னஞ்சல் அனுப்பவும்.

இந்தத் திட்டத்தின் மூலம் பணம் சம்பாதிப்பவர்கள் யார்?

யாருமில்லை.

இந்த வலைத்தளம் முழுக்க முழுக்க தன்னார்வலர்களால் செயல்படுகின்ற ஒரு வலைத்தளம் ஆகும். இதன் ஒரே நோக்கம் என்னவெனில் தமிழில் நிறைய மின்புத்தகங்களை உருவாக்குவதும், அவற்றை இலவசமாக பயனர்களுக்கு வழங்குவதுமே ஆகும்.

மேலும் இவ்வாறு உருவாக்கப்பட்ட மின்புத்தகங்கள், ebook reader ஏற்றுக்கொள்ளும் வடிவமைப்பில் அமையும்.

இத்திட்டத்தால் பதிப்புகளை எழுதிக்கொடுக்கும் ஆசிரியர்/பதிவருக்கு என்ன லாபம்?

ஆசிரியர்/பதிவர்கள் இத்திட்டத்தின் மூலம் எந்தவிதமான தொகையும் பெறப்போவதில்லை. ஏனெனில், அவர்கள் புதிதாக இதற்கென்று எந்தஒரு பதிவையும் எழுதித்தரப்போவதில்லை.

ஏற்கனவே அவர்கள் எழுதி வெளியிட்டிருக்கும் பதிவுகளை எடுத்துத்தான் நாம் மின்புத்தகமாக வெளியிடப்போகிறோம்.

அதாவது அவரவர்களின் வலைதளத்தில் இந்தப் பதிவுகள் அனைத்தும் இலவசமாகவே கிடைக்கப்பெற்றாலும், அவற்றையெல்லாம் ஒன்றாகத் தொகுத்து ebook reader போன்ற கருவிகளில் படிக்கும் விதத்தில் மாற்றித் தரும் வேலையை இந்தத் திட்டம் செய்கிறது.

தற்போது மக்கள் பெரிய அளவில் tablets மற்றும் ebook readers போன்ற கருவிகளை நாடிச் செல்வதால் அவர்களை நெருங்குவதற்கு இது ஒரு நல்ல வாய்ப்பாக அமையும்.

நகல் எடுப்பதை அனுமதிக்கும் வலைதளங்கள் ஏதேனும் தமிழில் உள்ளதா?

உள்ளது.

பின்வரும் தமிழில் உள்ள வலைதளங்கள் நகல்

எடுப்பதினை அனுமதிக்கின்றன.

1. www.vinavu.com

2. www.badriseshadri.in

3. http://maattru.com

4. kaniyam.com

5. blog.ravidreams.net

எவ்வாறு ஓர் எழுத்தாளரிடம் **Creative Commons** உரிமத்தின் கீழ் அவரது படைப்புகளை வெளியிடுமாறு கூறுவது**?**

இதற்கு பின்வருமாறு ஒரு மின்னஞ்சலை அனுப்ப வேண்டும்.

<துவக்கம்>

உங்களது வலைத்தளம் அருமை [வலைதளத்தின் பெயர்].

தற்போது படிப்பதற்கு உபயோகப்படும் கருவிகளாக Mobiles மற்றும் பல்வேறு கையிருப்புக் கருவிகளின் எண்ணிக்கை அதிகரித்து வந்துள்ளது.

இந்நிலையில் நாங்கள் http://www.FreeTamilEbooks.com எனும் வலைதளத்தில், பல்வேறு தமிழ் மின்புத்தகங்களை வெவ்வேறு துறைகளின் கீழ் சேகரிப்பதற்கான ஒரு புதிய திட்டத்தில் ஈடுபட்டுள்ளோம்.

இங்கு சேகரிக்கப்படும் மின்புத்தகங்கள் பல்வேறு கணிணிக் கருவிகளான Desktop,ebook readers like kindl, nook, mobiles, tablets with android, iOS போன்றவற்றில் படிக்கும் வண்ணம் அமையும். அதாவது இத்தகைய கருவிகள் support செய்யும் odt, pdf, ebub, azw போன்ற வடிவமைப்பில் புத்தகங்கள் அமையும்.

இதற்காக நாங்கள் உங்களது வலைதளத்திலிருந்து

பதிவுகளை
பெற விரும்புகிறோம். இதன் மூலம் உங்களது பதிவுகள் உலகளவில் இருக்கும் வாசகர்களின் கருவிகளை நேரடியாகச் சென்றடையும்.

எனவே உங்களது வலைதளத்திலிருந்து பதிவுகளை பிரதியெடுப்பதற்கும் அவற்றை மின்புத்தகங்களாக மாற்றுவதற்கும் உங்களது அனுமதியை வேண்டுகிறோம்.

இவ்வாறு உருவாக்கப்பட்ட மின்புத்தகங்களில் கண்டிப்பாக ஆசிரியராக உங்களின் பெயரும் மற்றும் உங்களது வலைதள முகவரியும் இடம்பெறும். மேலும் இவை “Creative Commons” உரிமத்தின் கீழ் மட்டும்தான் வெளியிடப்படும் எனும் உறுதியையும் அளிக்கிறோம்.

http://creativecommons.org/licenses/

நீங்கள் எங்களை பின்வரும் முகவரிகளில் தொடர்பு கொள்ளலாம்.

e-mail : **freetamilebooksteam@gmail.com**
FB : https://www.facebook.com/FreeTamilEbooks

G +: https://plus.google.com/communities/108817760492177970948

நன்றி.

</முடிவு>

மேற்கூறியவாறு ஒரு மின்னஞ்சலை உங்களுக்குத் தெரிந்த அனைத்து எழுத்தாளர்களுக்கும் அனுப்பி அவர்களிடமிருந்து அனுமதியைப் பெறுங்கள்.

முடிந்தால் அவர்களையும் “Creative Commons License”-ஐ அவர்களுடைய வலைதளத்தில் பயன்படுத்தச் சொல்லுங்கள்.

கடைசியாக அவர்கள் உங்களுக்கு அனுமதி அளித்து

அனுப்பியிருக்கும்
மின்னஞ்சலை**freetamilebooksteam@gmail.com** எனும் முகவரிக்கு அனுப்பி வையுங்கள்.

ஓர் எழுத்தாளர் உங்களது உங்களது வேண்டுகோளை மறுக்கும் பட்சத்தில் என்ன செய்வது ?

அவர்களையும் அவர்களது படைப்புகளையும் அப்படியே விட்டுவிட வேண்டும்.

ஒருசிலருக்கு அவர்களுடைய சொந்த முயற்சியில் மின்புத்தகம் தயாரிக்கும் எண்ணம்கூட இருக்கும். ஆகவே அவர்களை நாம் மீண்டும் மீண்டும் தொந்தரவு செய்யக் கூடாது.

அவர்களை அப்படியே விட்டுவிட்டு அடுத்தடுத்த எழுத்தாளர்களை நோக்கி நமது முயற்சியைத் தொடர வேண்டும்.

மின்புத்தகங்கள் எவ்வாறு அமைய வேண்டும்?

ஒவ்வொருவரது வலைத்தளத்திலும் குறைந்தபட்சம் நூற்றுக்கணக்கில் பதிவுகள் காணப்படும். அவை வகைப்படுத்தப்பட்டோ அல்லது வகைப்படுத்தப் படாமலோ இருக்கும்.

நாம் அவற்றையெல்லாம் ஒன்றாகத் திரட்டி ஒரு பொதுவான தலைப்பின்கீழ் வகைப்படுத்தி மின்புத்தகங்களாகத் தயாரிக்கலாம். அவ்வாறு வகைப்படுத்தப்படும் மின்புத்தகங்களை பகுதி-I பகுதி-II என்றும் கூட தனித்தனியே பிரித்துக் கொடுக்கலாம்.

தவிர்க்க வேண்டியவைகள் யாவை?

இனம், பாலியல் மற்றும் வன்முறை போன்றவற்றைத் தூண்டும் வகையான பதிவுகள் தவிர்க்கப்பட வேண்டும்.

எங்களைத் தொடர்பு கொள்வது எப்படி?

நீங்கள் பின்வரும் முகவரிகளில் எங்களைத் தொடர்பு கொள்ளலாம்.

- **email : freetamilebooksteam@gmail.com**
- Facebook: https://www.facebook.com/FreeTamilEbooks
- Google Plus: https://plus.google.com/communities/108817760492177970948

இத்திட்டத்தில் ஈடுபட்டுள்ளவர்கள் யார்?

- **Shrinivasan tshrinivasan@gmail.com**
- **Alagunambi Welkin** alagunambiwelkin@fsftn.org
- **Arun** arun@fsftn.org
- இரவி

Supported by

- Free Software Foundation TamilNadu, www.fsftn.org
- Yavarukkum Software Foundation http://www.yavarkkum.org/